11:21:05 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 06/19/91

Active

Project #: C-50-606 Cost share #: Rev #: 3
Center # : 10/24-6-R6839-0A1 Center shr #: OCA file #:
Work type : RES
Contract#: RESEARCH ASSISTANCE AGREEMENT Mod #: MEMO OF 6/14/91 Document : AGR
Prime #: Contract entity: GTRC

Subprojects ? : Y CFDA: N/A
Main project #: PE #: N/A

Project unit: VGU Unit code: 02.010.314
Project director(s):
HODGES L COMPUTING (404)894-8787

Sponsor/division names: TELEDYNE BROWN ENGINEERNG /
Sponsor/division codes: 219 / 005

Award period: 890815 to 910814 (performance) 910914 (reports)

Sponsor amount	New this change	Total to date
Contract value	0.00	80,000.00
Funded	0.00	80,000.00
Cost sharing amount		0.00

Does subcontracting plan apply ?: N

Title: CHARACTERIZATION AND GENERATION OF CAMOUFLAGE PATTERNS

PROJECT ADMINISTRATION DATA

OCA contact: Don S. Hasty 894-4820

Sponsor technical contact

JERRY C. EDWARDS
(205)726-1000

TELEDYNE BROWN ENGINEERING
300 RESEARCH PARK, P.O. BOX 070007
HUNTSVILLE, AL 35807-7007

Sponsor issuing office

CLYDE L. WARE, M/S 116
(205)726-1374

TELEDYNE BROWN ENGINEERING
300 RESEARCH PARK, P.O. BOX 070007
HUNTSVILLE, AL 35807-7007

Security class (U,C,S,TS) : U ONR resident rep. is ACO (Y/N): N
Defense priority rating : N/A N/A supplemental sheet
Equipment title vests with: Sponsor X GIT
NONE PROPOSED
Administrative comments -
REVISION ISSUED TO CHANGE PROJECT NO. FROM C-36-626 TO C-50-606; CENTER NO. CHANGED FROM R6839-0A0 TO R6839-0A1; UNIT CODE CHANGED TO VGU.

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

Closeout Notice Date 09/10/93

Project No. C-50-606 Center No. 10/24-6-R6839-0A1

Project Director HODGES L School/Lab GVU

Sponsor TELEDYNE BROWN ENGINEERNG/

Contract/Grant No. RESEARCH ASSISTANCE AGREEMENT Contract Entity GTRC

Prime Contract No.

Title CHARACTERIZATION AND GENERATION OF CAMOUFLAGE PATTERNS

Effective Completion Date 910814 (Performance) 910914 (Reports)

Closeout Actions Required:	Y/N	Date Submitted
Final Invoice or Copy of Final Invoice	Y	910919
Final Report of Inventions and/or Subcontracts	N	
Government Property Inventory & Related Certificate	N	
Classified Material Certificate	N	
Release and Assignment	N	
Other	N	

Comments EFFECTIVE DATE 8-15-89. CONTRACT VALUE $80,000.

Subproject Under Main Project No.

Continues Project No.

Distribution Required:

Project Director	Y
Administrative Network Representative	Y
GTRI Accounting/Grants and Contracts	Y
Procurement/Supply Services	Y
Research Property Managment	Y
Research Security Services	N
Reports Coordinator (OCA)	Y
GTRC	Y
Project File	Y
Other CARL BAXTER-FMD	Y
FRED CAIN-OOD	Y

Georgia Tech

Georgia Institute of Technology
School of Information and Computer Science
Atlanta, Georgia 30332-0280
404 · 894 · 3152

November 14, 1989

Jerry Edwards
Teledyne Brown Engineering
Cummings Research Park
P. O. Box 070007
Huntsville, Alabama 35807-7007

Dear Mr. Edwards:

As agreed upon during our meeting on Friday, this letter will serve as the monthly letter report for project G-36-626, *Characterization and Generation of Camouflage Patterns*, for the two-month period September 15 through November 15, 1989. Activity on this project during this period includes:

°Two meetings (9/22/89 and 11/10/89) with Jerry Edwards during which Mr. Edwards outlined the scope and goals of the project and gave an overview of the current "state-of-the-art" in camouflage manufacturing and evaluation.

°Two meetings with Dr. Ted Doll of GTRI to determine how the camouflage project that he is currently working on is related to our work.

°We have set up subbudgets to support a month of Dr. Sheffer's time and to fund a 1/3 time research assistantship for Mr. Akleman.

°We have taken data on a floppy disk that was digitized by Teledyne-Brown and ported it for display on a Silicon Graphics 4D workstation at Georgia Tech.

°We have reviewed the photographs of forest scenes that were provided by Mr. Edwards. We believe that the best scenes for preliminary analysis are: Open Pine - 1, Open Pine - 2, Open Pine - 3, Pine Trees - 1, and Pine Trees - 3.

Sincerely,

Larry F. Hodges, Ph.D.

Georgia Tech

Georgia Institute of Technology
School of Information and Computer Science
Atlanta, Georgia 30332-0280
404 · 894 · 3152

December 15, 1989

Jerry Edwards
Teledyne Brown Engineering
Cummings Research Park
P. O. Box 070007
Huntsville, Alabama 35807-7007

Dear Mr. Edwards:

This is our monthly letter report outlining our efforts on on the project: *Characterization and Generation of Camouflage Patterns* for the period 11/16/89 through 12/15/89. Our primary thrust this month has been to write software on the SGI graphics workstation to implement some basic image processing techniques so that we can examine their usefulness in identifying and characterizing basic patterns in an image of a natural scene. This preliminary analysis has including the following methods:

°Quantization of the image into four intensity levels.
°Histogram equalization of the gray levels in the image.
°Analysis of the image by filtering it through a highpass filter. Filtering was done using both a point template and a gradient (sobel) template.

We have not yet determined if any of these techniques will be useful in developing quantitative measures of images that will be useful in camouflage. During the next month we plan to continue the image processing approach. We will also make a video tape of the effects the filters have on the images that we will send to you with our next report.

Sincerely,

Larry F. Hodges, Ph.D.

Georgia Tech

Georgia Institute of Technology
School of Information and Computer Science
Atlanta, Georgia 30332-0280
404 · 894 · 3152

January 15, 1990

Jerry Edwards
Teledyne Brown Engineering
Cummings Research Park
P. O. Box 070007
Huntsville, Alabama 35807-7007

Dear Mr. Edwards:

This is our monthly letter report outlining our efforts on on the project: *Characterization and Generation of Camouflage Patterns* for the period 12/16/89 through 1/15/90. This report consists of a video tape illustrating the patterns that are generated from a basic scene using two different image processing filters and a script that accompanies the tape. The tape will be sent directly to you instead of being routed through OCA.

Sincerely,

Larry F. Hodges, Ph.D.

Video: TREES #1

Image 1: Scene created from the original data format supplied by Teledyne Brown.

Image 2: Scene after data has been reformatted for SGI *ipaste* command format.

Image 3: Histogram of the original image. In this image it is clear that the data is not equally distributed over the entire range of gray values. (Area in white indicates the distribution for grayscale values 0-255.)

Image 4: Image after processing with a histogram equalization routine to get equally distributed intensity data.

The eye responds logarithmically to intensity changes. We chose quantization levels logarithmic in base 2: 16,8,4,2. Since three colors or intensities are common in camouflage, we also looked at 3 quantization levels. Quantization is implemented by equal distances between gray levels (i.e., if there are 4 levels, the threshold values are 512/4 = 128, 128*2 = 256, 128*3 = 384 and 512. Histogram equalization guarantees that there will be approximately the same number of pixels for each gray level.

Image 5: 16 levels quantization

Image 6: 8 levels quantization

Image 7: 4 levels quantization

Image 8: 3 levels quantization

Image 9: 2 levels quantization

In the following images we compared two templates: gradient (sobel) and point in an effort to analytically define patterns that are characteristic of specific kinds of foliage. The gradient template emphasizes edges while the point template creates points.

Images 10-11: Gradient template filtering of 16 levels quantized image for two different threshold levels.

Image 12: Point template filtering of 16 levels quantized image.

Images 13-14: Gradient template filtering of 8 levels quantized image for two different threshold levels.

Image 15: Point template filtering of 8 levels quantized image.

Images 16-17: Gradient template filtering of 4 levels quantized image for two different threshold levels.

Image 18: Point template filtering of 4 levels quantized image.

Images 19-20: Gradient template filtering of 3 levels quantized image for two different threshold levels.

Image 21: Point template filtering of 3 levels quantized image.

Georgia Tech

Georgia Institute of Technology
School of Information and Computer Science
Atlanta, Georgia 30332-0280
404•894•3152
Fax: 404•853•9378

February 15, 1990

Jerry Edwards
Teledyne Brown Engineering
Cummings Research Park
P. O. Box 070007
Huntsville, Alabama 35807-7007

Dear Mr. Edwards:

This is our monthly letter report outlining our efforts on on the project: *Characterization and Generation of Camouflage Patterns* for the period 1/1/90 through 1/31/90. We are still exploring image processing techniques to characterize the basic colors and patterns in a scene. Once we have developed a process that will extract characteristic shapes from a scene, we can use a fractal generation algorithm to produce camouflage patterns. Ergun thinks that a pyramid scheme described by P.J. Burt in Multiresolution Image Processing and Analysis (A. Rosenfeld, Ed.) can be used to do this. By repeatedly reducing the size of a scene through appropriate filters we can produce intensites for camouflage patterns. By reversing the process (expansion) we think we can identify the patterns that are characteristic of the scene. We are currently identifying properties that this expansion process must exhibit.

Sincerely,

Larry F. Hodges, Ph.D.

CHARACTERIZATION AND GENERATION OF CAMOUFLAGE PATTERNS

Monthly Report

Submitted To

Jerry Edwards
Teledyne Brown Engineering
Cummings Research Park
P.O. Box 070007
Huntsville, Alabama 35807-7007

By

Ergun Akleman, Larry Hodges and Albert Sheffer
Georgia Institute of Technology
Atlanta, GA 30332

March 15, 1990

CHARACTERIZATION AND GENERATION OF CAMOUFLAGE PATTERNS
February report, 1990

Ergun Akleman, Larry Hodges and Albert Sheffer
Georgia Institute of Technology
Atlanta, GA 30332

1. Introduction.

The problem we are researching is to design a camouflage pattern for a given natural background. Defining a background is difficult. Natural scenes change with parameters such as location of the sun, direction and power of the wind, and weather. Since the background is dynamic a definition of its characteristics will never be precise. In this report we will assume a static background.

This assumption will enable us to use a photograph as a meaningful replica of a given background. We note, however, that in a photograph some information about the background scene it represents is incorrect. The incorrect information must be corrected to produce the desired camouflage image. Sources of incorrect information in the photograph include the following.

1) Photographic film does not reproduce the natural colors in the scene faithfully, therefore the color of the objects in the background scene must be analyzed independently.

2) The intensity in the photograph does not correspond to the actual intensity of the background, since the former completely depends on the camera exposure at the time the photograph is taken. Independently from the photograph, the overall and local intensities in the background scene must be found using a lightmeter. These intensities help to adjust gray levels.

3) If the reflection properties of the camouflage material are different from that of the background, the overall intensity of the camouflage must be adjusted accordingly. This is a particular problem with net material.

We will work from a photograph assuming that these corrections can be made. In section two of this report we explain the pyramid data structure and its relationship to the human visual system. In the third section we give ideas for an expansion filter. In the fourth section we explain how we can generate an expansion filter.

2. Pyramid Data Structure

Assume that we take pictures of a painting. While we increase the distance between our camera and the painting, the picture of the painting will be smaller. The details become indistinct. If we continue to increase the distance the picture will close to a point. If we stack all these images according to the distance between the camera and the painting we will get a pyramid structure. A pyramid data structure is given figure 1. Such a structure can be used to support multi-resolution detail in a scene[1].

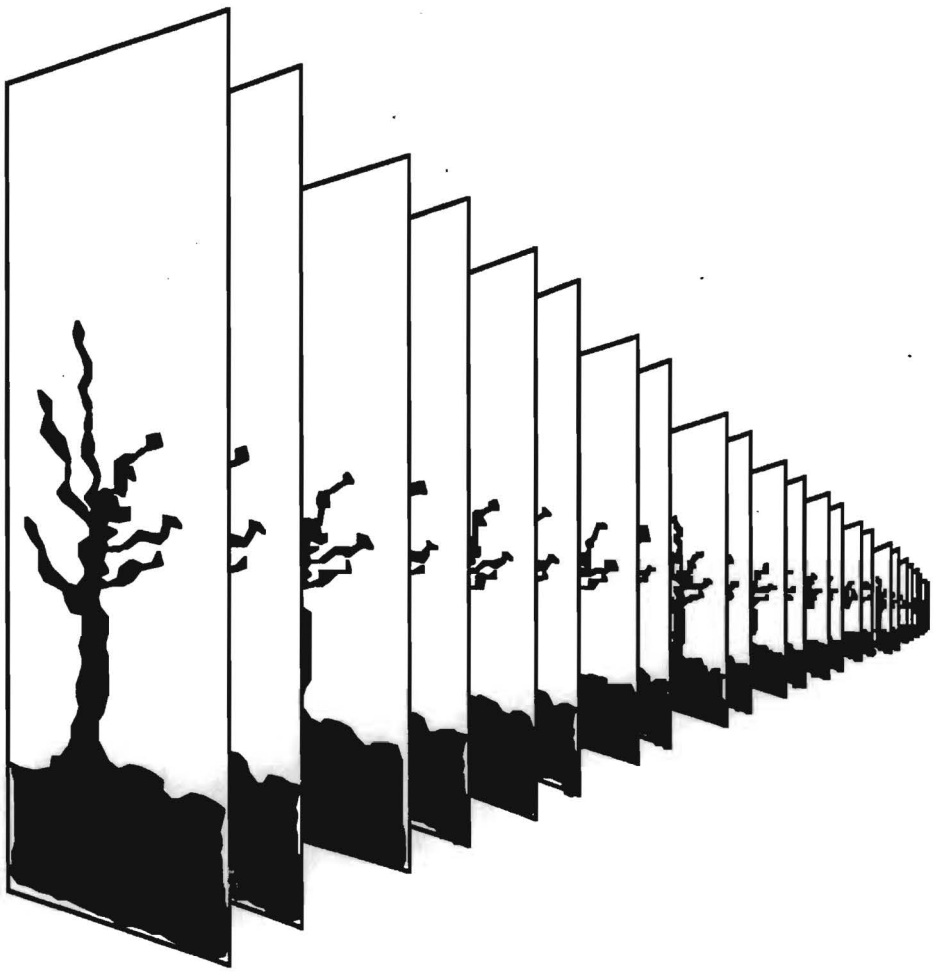


Figure 1.

The procedure described above can not be used for dynamic images. Unfortunately natural scenes, which we are interested in as background, are dynamic, they will change with time. Moreover, it is very difficult task to change the distance without changing the camera angle. Therefore instead of having to take a set of photographs we would like to generate the whole pyramid from only one photograph.

Generally, the photograph is much smaller than natural scene it represents. Thus, the position of the photograph on the pyramid will not be the base image but will appear somewhere in the middle. If we would like to find all the images in the pyramid we should create two different types of mappings. Contraction mappings that will create images smaller than the photograph, and expansion mappings that will create the larger images. The position of the photograph in pyramid is given in figure 2.

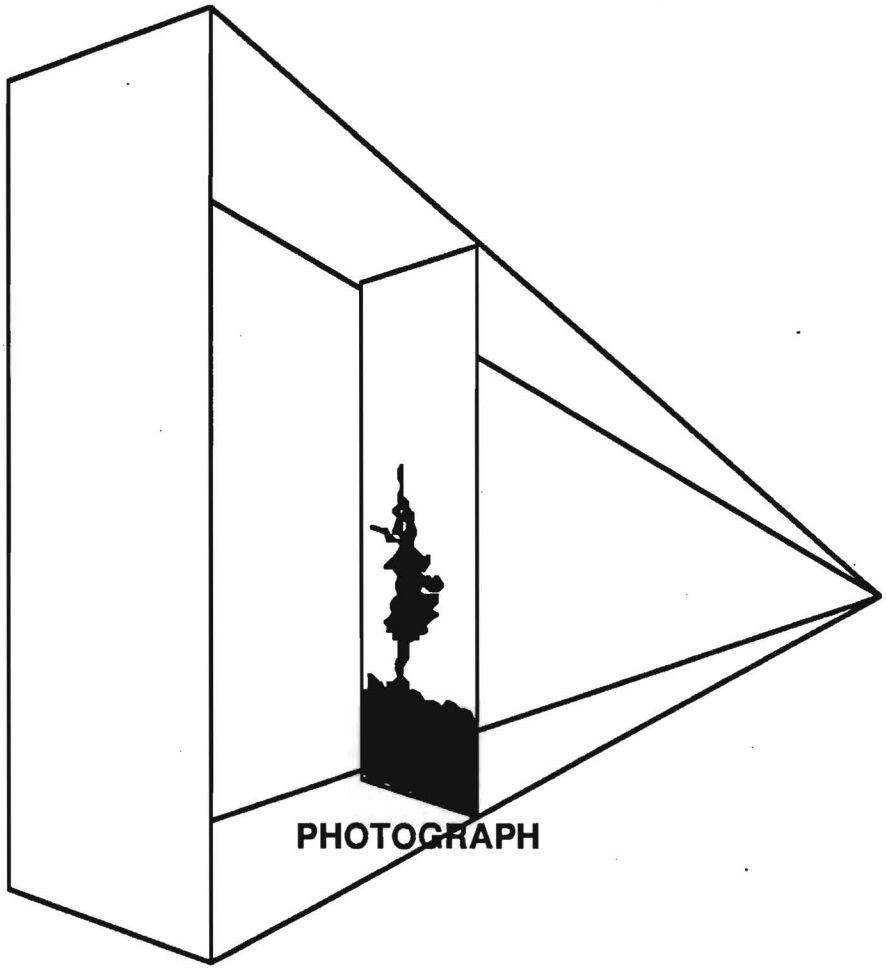


Figure 2.

If we consider a pin hole camera model, we can determine the characteristics of the contraction mappings we should use. The model is given in the figure 3. From projective geometry the length of the picture, E, depends on the size of actual image by the given equation $E = (d_0 \cdot I) / d$.

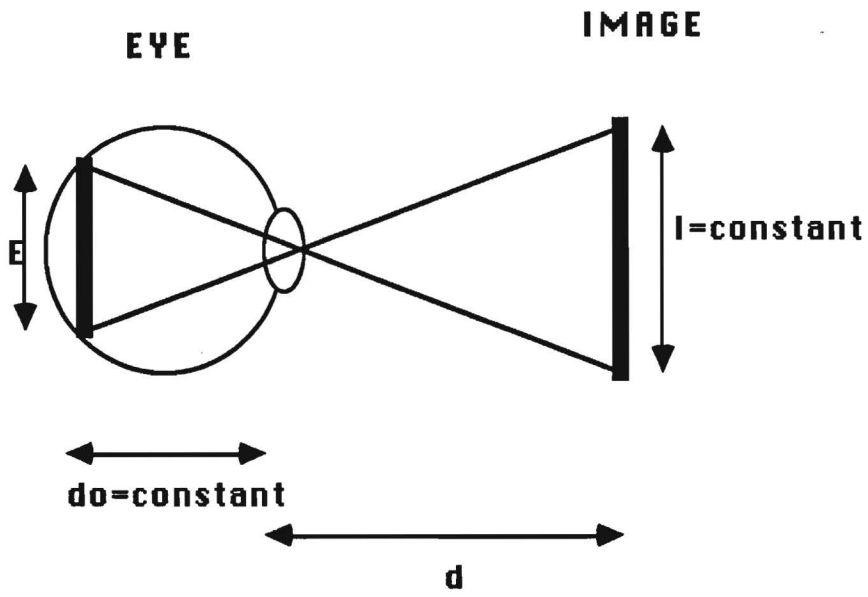


Figure 3.

We can think of the back of the human eye as consisting of receptors (rods and cones) that determine hue, saturation and intensity of color. If we look at one receptor we can see that both a camera and the human eye work as a lowpass filter. This phenomena is explained in the figures 4 and 5. For a longer distance the same point will be interpreted darker by a particular receptor. If we consider the whole set of receptors, we conclude that with distance the image not only becomes smaller but is also filtered.

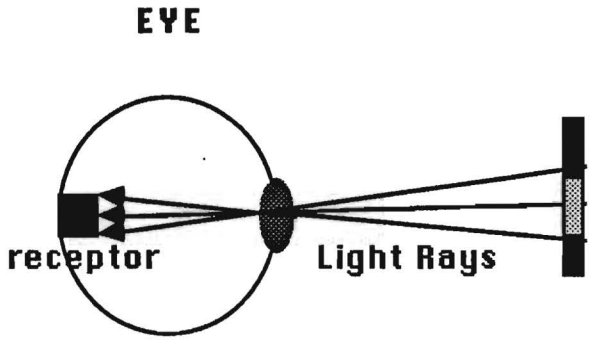


Figure 4.

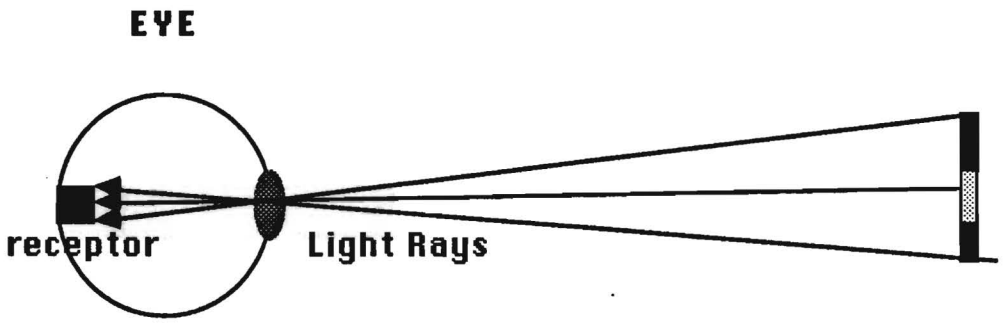


Figure 5.

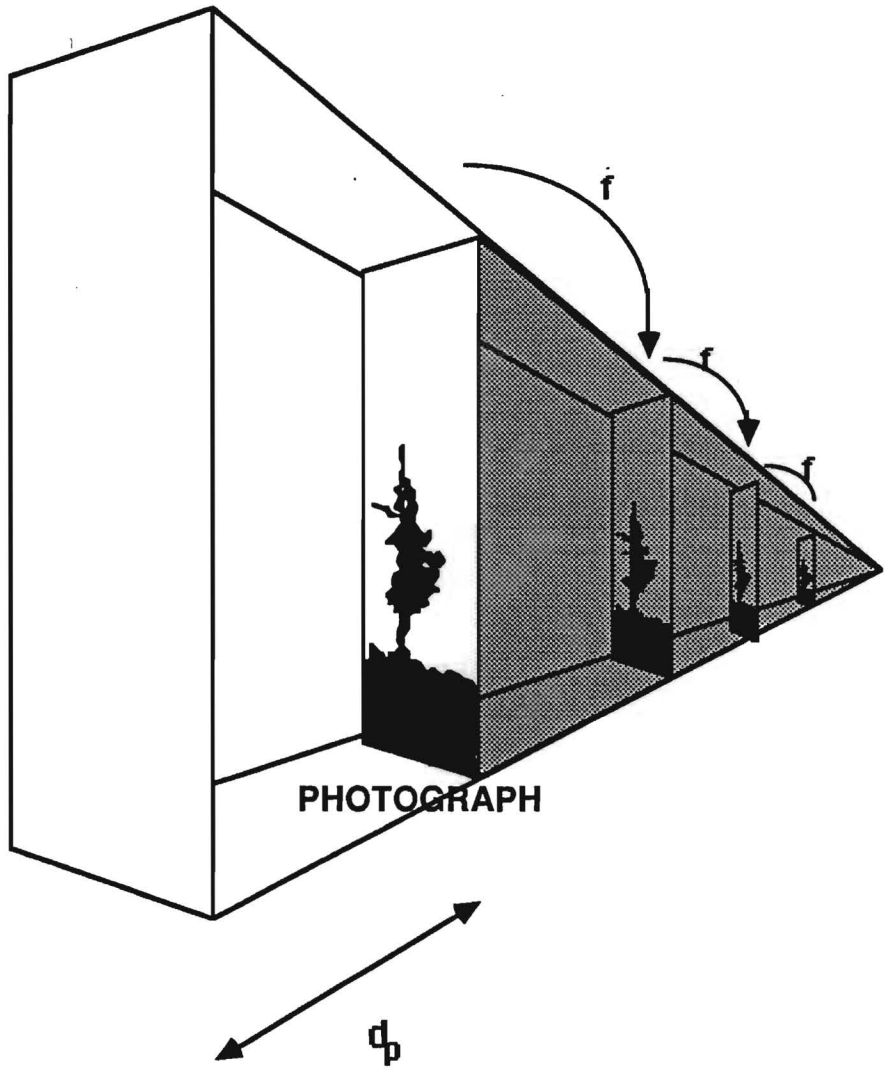


Figure 6.

3. Expansion Filter

Starting from the photograph we will also generate the upper section of the pyramid based on the same contraction mappings. The construction is shown in figure 6. In the figure, f is a contraction mapping. Assuming that the photograph image is P, we will get a set of images { P, f(P), f^2(P), f^3(P),... f^n(P)} which is the upper section of pyramid.

Using the same idea we conclude that the lower part of the pyramid can be constructed using the inverse of the contraction mapping as shown in figure 7. The whole pyramid will be { f^{-m}(P),.... f^{-2}(P), f^{-1}(P), P, f(P), f^2(P),.... f^n(P)} where f^{-m}(P) corresponds to the base of the pyramid or to the camouflage image. Using an inverse function we guarantee that the pattern produced will be an exact replica of the photograph. An important observation is that the inverse mapping is not unique.

Instead of mimicking the photograph exactly we can loosen our restrictions slightly and propose an expansion mapping R such that the error in the upper region is within some specified tolerance. Then the pyramid is represented by { R^m(P),.... R(P), R(P), P, f(P), f^2(P),.... f^n(P)}. Since the original image can be written as $f^m(R^m(P))$ the error will be given as

$$d(f^m (R^m(P)), P) < E$$
$$d(f^{m+1}(R^m(P)),f(P)) < E$$
$$d(f^{m+2}(R^m(P)),f^2(P)) < E$$
$$\vdots$$
$$d(f^{m+n}(R^m(P)),f^n(P)) < E$$

where d(A,B) is a given distance between two images A and B and E is the tolerated error. In this work we will concentrate on the inverse mapping of f.

4. Generation of Inverse mapping

Using the human eye or pinhole camera model we choose to use a very simple generalized lowpass filter. For a generalized filter the algorithm is

1. take a neighborhood of pixels
2. add gray values of pixels
3. divide it to number of pixels in the neighborhood

4. replace this neighborhood with a pixel with gray value as found in 3.

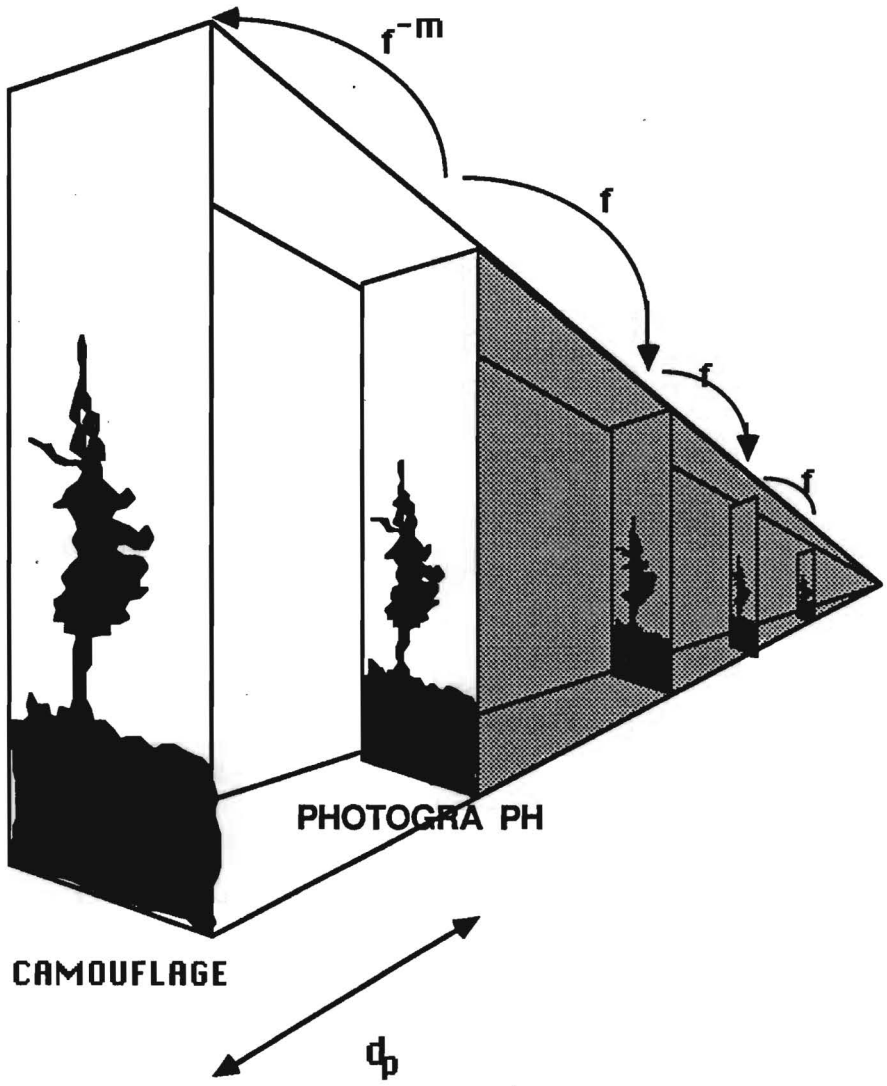


Figure 7.

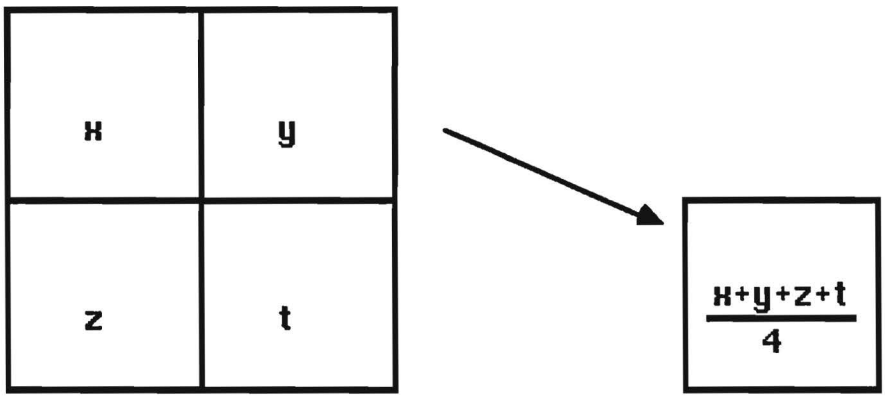


Figure 8.

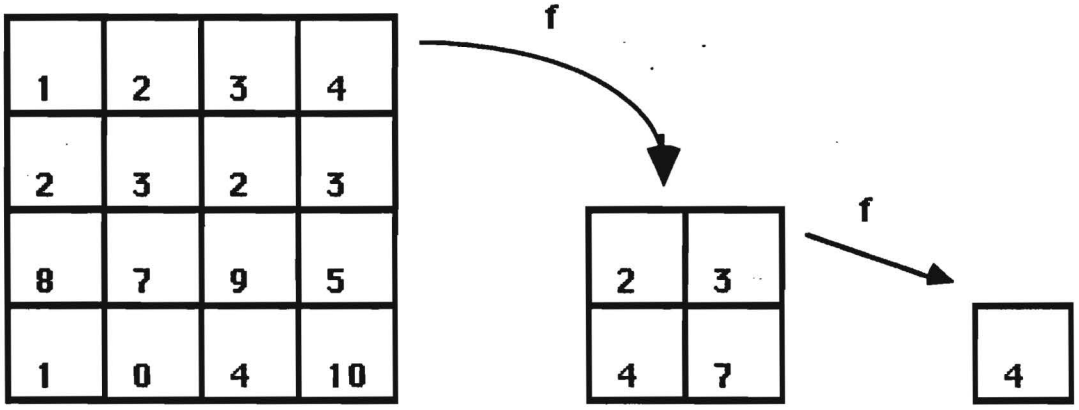


Figure 9.

We will concentrate on a 2x2 neighborhood since it can easily be generalized to an nxn neighborhood. If the gray values of this neighborhood are x,y,z,t then the gray value of the new pixel, ***a***,will be a= (x+y+z+t)/4. The filter is shown in figure 8 and an example is given in figure 9. The inverse mapping, shown in figure 10, for a given gray value, should find a 2x2 subimage with gray values of {x,y,z,t} such that the formula above holds. The mapping is not unique. If x,y,z,t is a solution, {x-1, y+1,z, t} is also a solution since (x-1)+(y+1)+z+t=x+y+z+t. In addition, the order of {x,y,z,t} is not important.

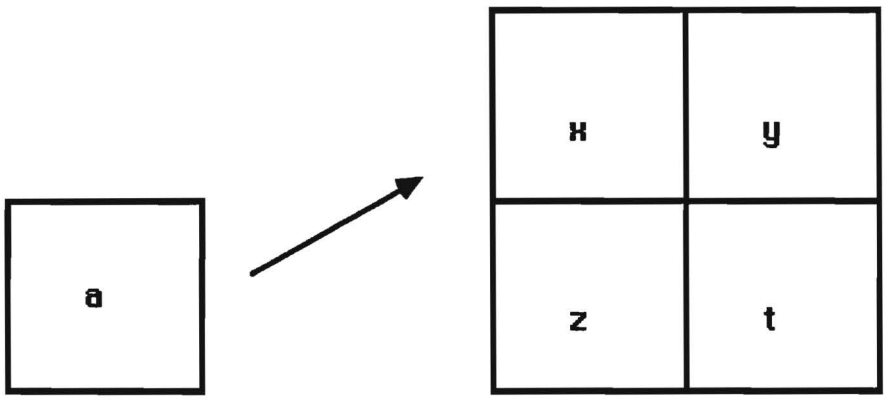


Figure 10.

To make production inexpensive we should use as few different gray values as possible. This constraint leads us to find a minimum set of gray values $\{x_1,x_2,x_3,....x_N\}$ such that for any given gray value of *a* from a photograph, there will be four gray values from this set of x's such that

$a = (x_i + x_j + x_k + x_l)/4$, i,j,k,l can be any value between 1 to N.

There is no optimum solution to this problem. Moreover, exact solutions are not very useful. Therefore we look for an approximate solution.

If the photograph has all 512 gray values, then 17 gray values are enough to get the approximately same impression of the photograph with some missing gray values. The set of gray values is{0, 4, 20, 40, 72, 108, 156, 208, 250, 296, 356, 400, 440, 468, 492, 504, 512 }.Since the actual number of gray levels in a digitized photograph is generally much less than 512 it is possible to reduce the number considerably.

If we use 10x10 neighborhood in filter it is possible to represent the whole range of 512 gray levels with just 4 gray levels such as {0,100,380,512}. In an actual scene the range between the natural scene and a photograph is more than 10x10. Therefore the approach is very promising.

Even more reduction can be made by assigning black and white blocks to every gray values as is shown in figure 11.

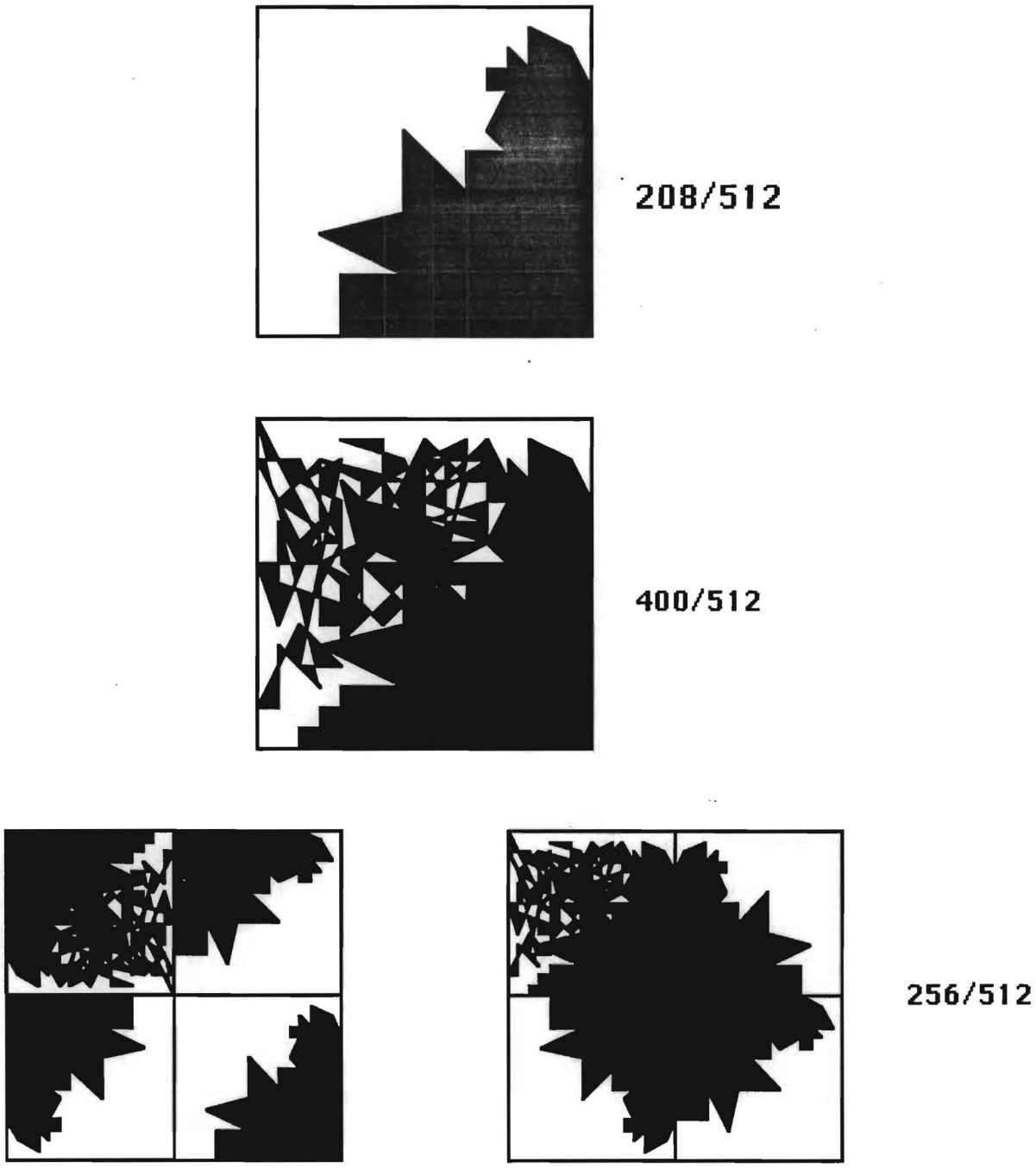


Figure 11.

6. Summary

In this report we suggest a general technique to support a multiresolution approach to the analysis of background scenes. We believe that the basic pyramid data structure described can be used to develop software tools for generation of camouflage patterns.

7. References

1. P. J. Burt "The pyramid as a Structure for Efficient Calculation", in Multiresolution Image Processing and Analysis, A. Rosenfels, Ed. (Springer Verlag, Berlin, 1984) pp. 6-35.

OVERVIEW OF

CAMOUFLAGE GENERATING

SOFTWARE FUNCTIONS

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

Larry F. Hodges
Ergun Akleman

Georgia Institute of Technology
Atlanta, Georgia

April 15, 1990

OVERVIEW OF CAMOUFLAGE GENERATING SOFTWARE FUNCTIONS

INTRODUCTION

In this report we give an overview of camouflage generating software based on the ideas presented in our last report. To generate camouflage patterns we will use a section of a photograph of a scene. Starting from this section using extension filters, the user will generate a camouflage image. As we presented in the earlier report, extension filters are not unique. Therefore from any photographic image, a user can generate different camouflage patterns. The issue discussed here is which options should be provided to the user.

We will assume that a digitized version of a photograph is provided. Since the user may not want to base the camouflage pattern on the entire image, he must be given the option of choosing a window from the digitized image. Once the window is selected, the user may also choose to operate from a filtered version of the image. Thus the software must have the ability to low-pass filter the image. (As explained in the earlier report, low-pass filtering alters the image so that it appears to be viewed from a further distance.

To generate the camouflage pattern the user will exchange every pixel in the image by a square image block as shown in figure 1. The process of camouflaging is based on preservation of visual effect. Henceforth, when the image block is observed from distance, it will give the same visual effect as the pixel. The average gray value of the image block will be equal to the pixel's gray value.

In this process any image block which gives the same color effect as that of the pixel from a certain distance can be used. Since the expansion filter is not unique, it is possible to give to the user a large number of options to choose from. The most important thing in an image block are the image

elements used. For example, all the image blocks can be composed only of lines if the user wishes to camouflage a pine tree. To generate an image block there are basically two parameters. The first is the image elements to be used in the block and the second is the color of the elements. For example, figure 1 consists of black polygons and gray ellipses.

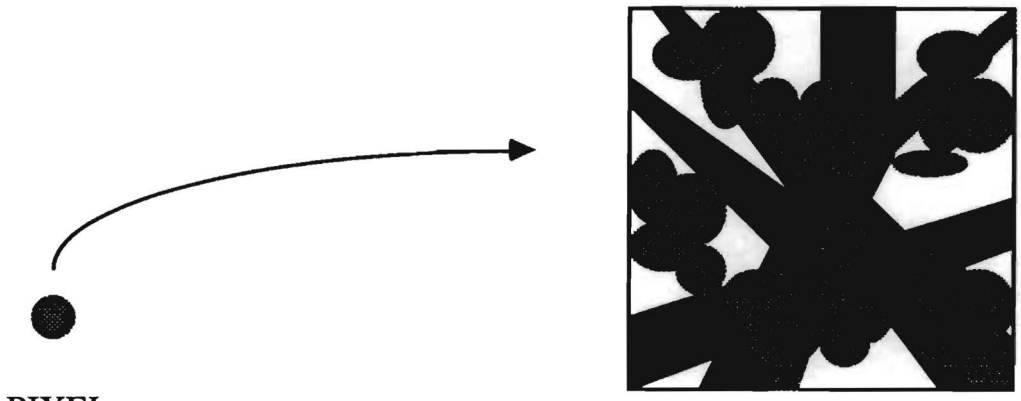


Figure 1.

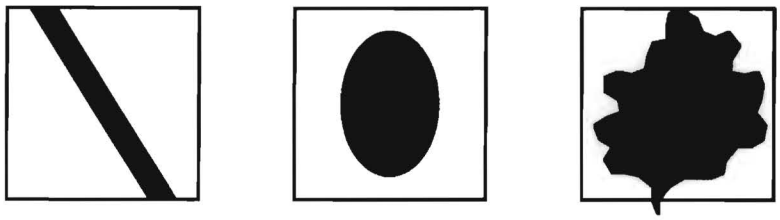
After the user chooses the image elements, he/she must also choose a production method to generate image blocks related to every pixel. In the following section we discuss the image elements and color needed to generate camouflage. The last section will explain choice of methods for generation of the image blocks that will replace every pixel.

IMAGE ELEMENTS AND COLOR

The image elements in the blocks must be related to the environment which is to be mimicked. For example, to make a camouflage image for sand, a collection of particles might be the best choice. Generally speaking, the software should provide a menu of basic image elements to choose from. These could be basic shapes such as lines, circles, ellipses, and polygons. Some possible image elements to choose from are shown in figure 2.

In a block there might be any collection of these image elements. Assuming that image elements can only be black and the background of

the block is white, the ratio of the covered area by shades to the area of the block will give the gray value of this particular block.



SOME IMAGE ELEMENTS

Figure 2.

The image elements can have any color. Lets assume the user wishes to generate a tree. The user will decide not only the image elements but also color for every particular image element. Figure 3 shows the same image element occurring in three different colors (shown here as three different gray values.)

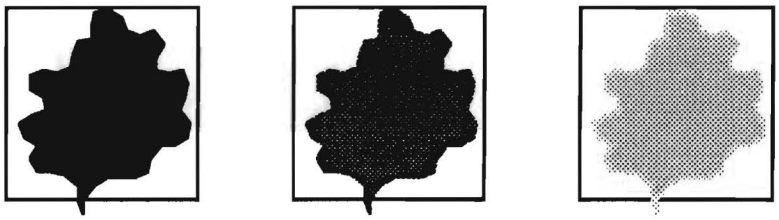


IMAGE ELEMENT WITH COLOR

Figure 3

GENERATION OF IMAGE BLOCKS

The user must control the generation of image blocks. After choosing the color and image elements, he should order how to put them together to match with the pixels. The average color of the image block is not affected by the position of the elements in it. As a result we can put an image element anywhere we want in an image block. If the user would like to get

the camouflage effect for short distances, the choice of generation process becomes important. Fractal generation might be a choice, using not only the image elements but also their smaller versions. Another choice might be a probabilistic approach. The position of any image element can be chosen randomly.

Exchanging every pixel by an image block can be done several ways. These ways should be provided to the user also. The first approach is a static approach. There is an image block associated with every color value, and every pixel according to its color value is changed to the related image block. The second approach is dynamic. Every image block is generated just for that particular image. In this approach there might not be any repetition in the generated camouflage pattern. We could also take a hybrid approach. For every color value there could be several choices of image blocks. One image block is then chosen randomly from among them. If the number of choices of image blocks is large, then this method will be functionally the same as the dynamic method.

SUMMARY

In this report we have discussed ideas for basic options of a camouflage generator program. Eventually we anticipate that implementation of this software will require a combination of image processing and either fractal or probabilistic image generation techniques. Further work must also be done in defining a user-interface for the camouflage generator.

BASIC EXPANSION FILTER DESIGN

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

MAY 15, 1990

Larry F. Hodges
Ergun Akleman

BASIC EXPANSION FILTER DESIGN

Introduction

An expansion filter algorithm is an integral part of the proposed camouflage software that we are designing. In this report we give an overview of the expansion filter we have developed. We have implemented a preliminary version of this algorithm in the C programming language on a Silicon Graphics 4D superworkstation.

This report gives a brief overview of the expansion filter algorithm and the fractal technique used to generate gray levels.

Expansion Filter

We expand an area in a scene by replacing every pixel by a square image block consisting of four or more pixels. Each image block is constructed using only a limited number of gray levels (usually 2, 3 or 4). Each image block is constructed so that its average gray level is equal to the gray level of the pixel that it replaces.

As described in our earlier reports, there is no unique process with which to construct an image block corresponding to particular gray value. In our implementation we use a random fractal generation to build the image block.

Fractal Generation of Image Blocks

As an example of fractal generation of image blocks we will look at the generation of a square using an algorithm based on Iterative Function System (IFS) theory[1,2]. The basic idea is shown in figure 1 .

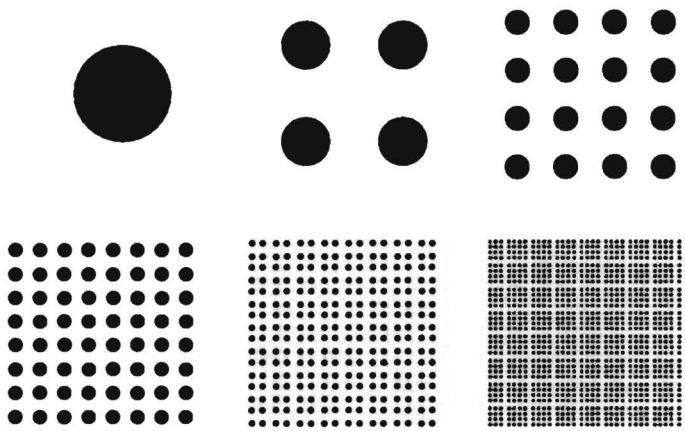


Figure 1. Fractal generation of a square

We begin with a basic shape and reduce its size by one half in both x and y directions. We then place copies of this reduced shape in the the upper left part of square, upper right part of square, lower right part of square and lower left part of square. If we iteratively continue this process for each of the reduced size shapes, every iteration gives us a better approximation to a solid square.

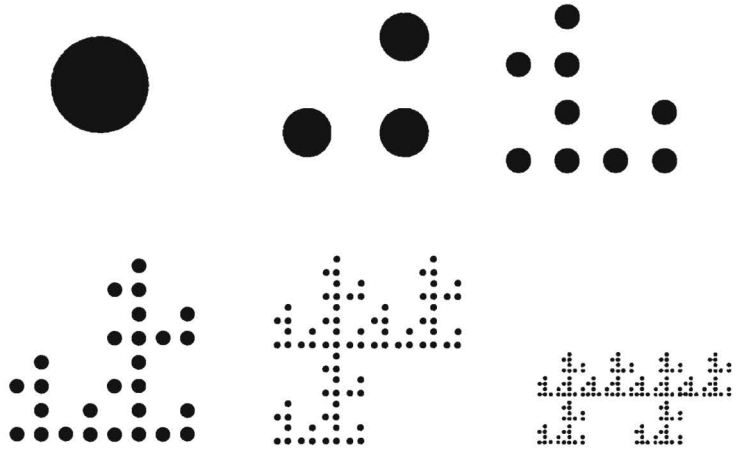


Figure 2. Fractal generation with alternating functions

In figure 2, the generating functions are the same, however we do not use all of them in each iteration. In every other iteration, the function that renders the smaller image in the upper left corner is deleted. In every third iteration the function that renders the smaller image in the upper

right corner is missing, and in every fifth iteration the function that renders object in the lower right corner is missing.

This generation will create seven different shapes according to the number of iterations implemented as shown in figure 3.

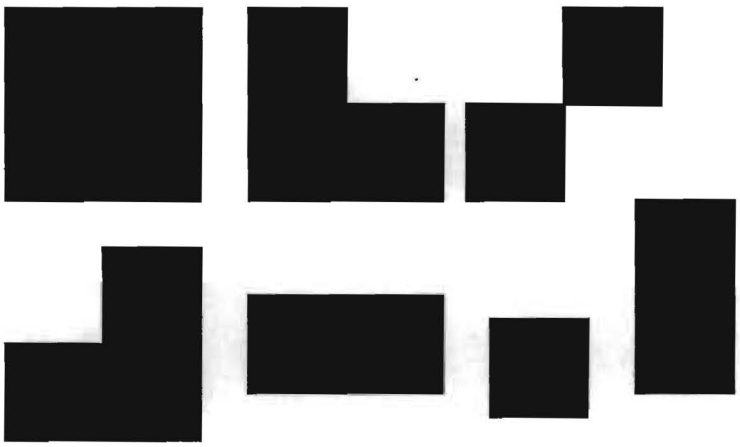


Figure 3. Basic shapes obtained by the procedure in figure 2.

According to IFS theory, we can randomize this technique and we can vary the gray level intensity used to generate the image. Based on this modification we can control the number of points on any part of the square to match the gray levels of the original image. In the figure 4, we illustrate the probabilistic algorithm for the probabilities in the image in figure 2.

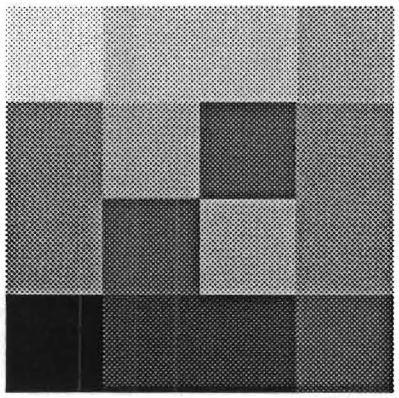


Figure 4. Probabilistic algorithm

The advantage of this algorithm is that it generates approximately smooth transitions from one image square to another while allowing us to expand the detail in an image. Gray values in each square always average the gray value of the pixel that it is derived from so that the original image is visible at a particular distance from the expanded image.

Although our preliminary implementation of this algorithm is written to generate square images, it can be modified eventually to include any kind of IFS image generation, such as leaves or branches.

References

1. Barnsley, M. F. and S. Demko, "Iterated Function Systems and the Global Construction of Fractals", Proceedings of Royal Society of London A399 (1985) 243-275.

2. Barnsley, M. F., Fractals Everywhere, Academic Press, San Diego, 1988.

AUTOMATIC GENERATION OF

FRACTAL IMAGES

MONTHLY REPORT

Prepared for

TELEDYNE BROWN ENGINEERING

May 15, 1990

Larry F. Hodges
Ergun Akleman
Theodore Doll

AUTOMATIC GENERATION OF FRACTAL IMAGES

We will assume that an image is provided and using our current software tools a square portion of the the image is chosen to be coded with a fractal approximation.

1. Fractal Dimension

We must first define a fractal dimension for an image block from which we wish to develop a fractal camouflage characterization. The standard procedure for experimentally determining fractal dimension for a two-color image (usually black and white) is as follows[1].

Cover the black and white image by disks of radius E for a range of E-values from an E value large enough to cover a square portion to an E value that is equal to the radius of one pixel. The number of disks necessary to cover the image for any given E, N(E), is related to its fractal dimension. For any two values of E, E_1 and E_2, one can compute a value related to fractal dimension using the equation:

(log(N(E1))-log(N(E1)))/(log(E1)-log(E1))

If the object is a true fractal this equation gives a true fractal dimension when E1 and E2 goes to zero with the behavior E1= delta + E2. Since a digitized image is not truly a fractal we can only find an approximate value but in most of the cases this approach gives satisfactory results.

2. Extended Approach for Images of more than Two Colors

To find an approximate fractal dimension for a multi-intensity image such as a digitized forest scene we must extend the basic fractal generation procedure. We are now testing the following procedure for grayscale images.

Start from the initial image portion and find its average intensity value, A. Divide the initial (square) image portion into smaller squares and find the average for every smaller square. If the average is smaller than A, then attach the value O otherwise 1. Find the sum of these numbers, take the logarithm and plot the result in logarithmic scale according to the size E of the smaller squares. There will be a portion in this plot such that it will behave approximately linearly. The slope of this portion will give approximate fractal dimension of the image.

3. Generation of Fractal Image

In the linear portion of the log scale choose any size E as a representative to generate iterated functions. Use the squares associated with this part of the log scale as maps to scale the whole image into the given boxes. Assume that for box $B_{i,j}$ the average value is given as $A_{i,j}$ then divide this value to A resulting $A/A_{i,j}$. Use this value as probability of the map given by the box $B_{i,j}$.

Assume that the coordinates of the initial portion is given by (x_1,y_1) and (x_2,y_2) where (x_1,y_1) represents the lower left corner of the initial square portion and (x_2,y_2) represents the upper right corner. Also assume that the square, $B_{i,j}$, is given by $(x_{i,j}1,y_{i,j}1)$ and $(x_{i,j}2,y_{i,j}2)$. The equation of the related map is given by the following equation:

$w(x)=x_{i,j}2*(x-x1)/(x2-x1)+x_{i,j}1$
$w(y)=y_{i,j}2*(y-y1)/(y2-y1)+y_{i,j}1$
with probability = $A_{i,j}/A$

Let there be n*n squares. Then there will be n*n maps to generate a fractal image. The most important aim computationally is to reduce the value of n. However larger values of n bring the fractal approximation closer to the original image.

4. References

1. Barnsley M. F. Fractals Everywhere, Academic Press, (1988).

Developments in the Image

Block Fractal Algorithm

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

June 15, 1990

Larry F. Hodges
Ergun Akleman

Developments in the Image Block Fractal Algorithm

Introduction

In the last report we described square block generating Iterated Function Systems (IFS). We believe that this basic approach is very promising for camouflage generation. Extensions to the basic idea must still be developed. In this report we described the extensions that we are currently working on.

The Probabilistic IFS Algorithm

As we described in the last report, IFS theory is based on a group of linear and contractive transformations. We apply these transformations to a two-dimensional set of points, and we get a new set of points. With iteration of the transformations the resulting sequence of sets converges to an particular image defined by those transformations.

Because of convergence property, it is possible to begin with a single data point instead of an entire set of points. It has been proven that the sequence of sets of points will converge to the same image.

Problems

The basic problem with this approach for camouflage generation is that the algorithm may not converge to the image (camouflage pattern) we desire. With the addition of a new point, we increase the gray level, until the average grey value reaches a desired intensity that matches that of the expanded pixel intensity. IFS theory does not guarantee that this value will be obtained. The algorithm could instead cycle through the same points.

As a result if we use the original algorithm, it might never converge to the proper total gray value. To increase the performance of the algorithm and to make sure the algorithm converges we propose several additional methods.

Additions

Our first concern is to make sure that the algorithm terminates and does not continually cycle through a series of points. Initially for this purpose we simply added a counter. If the number of iterations exceed a certain value the algorithm automatically stops. However, if the counter terminates the algorithm before the desire gray level is obtained, the resulting image does not fit our criteria for *good* camouflage. Since a group of pixels cannot then obtain their ideal average grey level, a noise is added to image. Therefore this method is not very desirable.

Our second idea is to use a dynamic starting image near to desired gray level instead of starting from minimum grey level. For example, if we assume that we have all ranges of grey levels and that we have two starting points one is completely white, the other one is half black and half white as it is shown in figure 1.

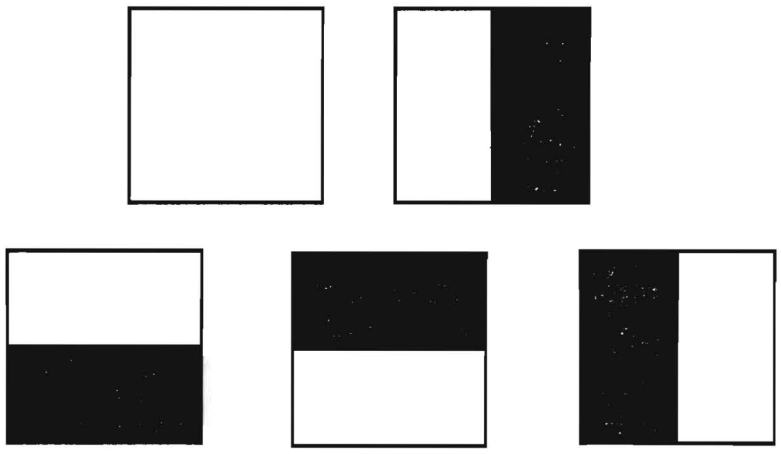


Figure 1.

Adding just one new starting image, we can use it in four different ways as in figure 1. These arrangements will be based on the grey values of

neighboring pixels. It is also possible to add quarter black and quarter white blocks, ect.

A third idea is to use subtraction instead of using addition of grey values. Instead of beginning the iteration with a white block and with every iteration increases the grey value, we begin from a black image, and add white points so as to decrease the grey value. In such a way we can limit the nonlinear distortion in darker regions.

Summary

We have examined three methods that increase the probability of convergence of the IFS algorithm to an acceptable value. It is probable that some combination of approaches will be necessary in a final algorithm.

Size and Color of Primitive Image Elements

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

July 15, 1990

Larry F. Hodges
Ergun Akleman

Size and Color of Primitive Image Elements

An image block in a scene consists of primitive image elements. In a raster graphics environment image elements are usually pixels on the display screen. However, for a camouflage image on fabric, the production constraints will force us to use larger primitives than a pixel. In this report we will describe the generation of primitive image elements for camouflage production.

We would like image elements to be small relative to the size of the image and the viewing distance of the observer. Currently we are requiring that the size of a image element must be smaller than one minute of arc. This approach provides us with a scale independent measure for basic patterns. If we assume X is the maximum length of one side of image element allowable. The area of this square will be X^2.

If production constraints restrict us to only two image colors, and we would like to generate 2^n different gray levels, we should divide this square into 2^n squares (The new primitive image elements are these squares.) Then the length, A, of a side of the new primitive image element is $X/2^{n-1}$.

For example, if we assume that we are viewing a camouflage pattern from 500 meters then X = 6 inches. If we would like to create 64 different shades of green using only two shades of green, A=6/32 inches.

There is a tradeoff between number of colors used in the pattern and the size of primitives. For example, if we use three colors instead of two, we can generate 2^n different gray levels by dividing a square into 2^{n-1} smaller squares. The length, A, of a side of the small squares is $X/2^{n-2}$. Addition of one basic color means that the side of the smallest image elements can be doubled and the area can be made four times larger (subject to the one degree of arc constraint).

In the example above, A becomes 6/16 inch=3/8 inch. These values (even A=6/32) seem reasonable for printing on fabric. If we assume that the pattern will be viewed through a 7X gunners sight, the size of a primitive element may become more critical. For example, if we assume 7x magnification of a camouflage pattern seen from 500 meters then X=1 inch, and if we would like to create 64 different shades of green using only two shades of green, A=1/32 inches. If we use three colors, A=1/16 inches. If reduce the number of grayshades from 64 to 32, we can make the side two times larger, A=1/8 inches.

Since the range of colors in current camouflage is small, we assume that 64 or 32 gray shades are sufficient. In the photographs with which we are currently working the range of grayshades is between 32 and 64.

Our intention is to design software such that the choice of size of primitive elements, number of colors, and number of grayshades can be determined by the person using the software.

IMAGES FROM PRELIMINARY SOFTWARE

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

August 15, 1990

Larry F. Hodges
Ergun Akleman

Images From Preliminary Software

This month's report will illustrate current progress with images generated with our preliminary software.

Image 1. The digitized photograph we used to generate a camouflage pattern is shown. We assumed that the photograph is taken from a distance of approximately 512 feet and the average height of the trees is approximately 64 feet. The scale of the photograph is 1/64th the size of the original scene.

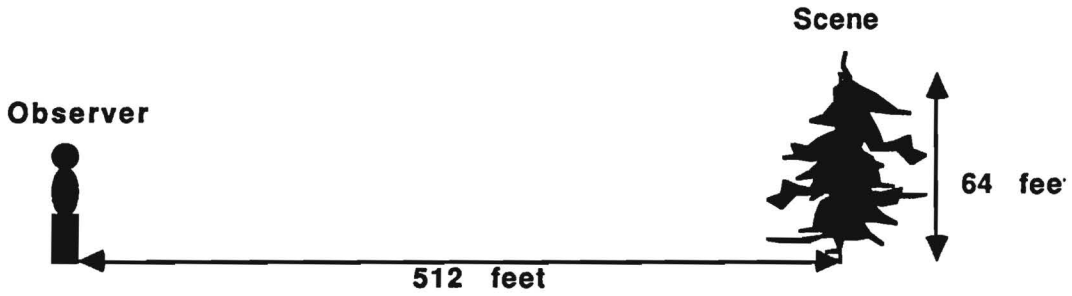


Figure 1. Assumptions on camera position and height of a tree

Image 2. From one of trees in the digitized photograph we choose a square patch measuring 0.125 feet per side. In this image we have replaced the original data with copies of the patch data in several different places. Note that even though the patch was taken from a specific part of the image, it still blendsalmost imperceptible into other parts of the scene.

Images 3, 4 and 5. These three images illustrate the detail produced by our expansion algorithm for camouflage patterns. All three images were based on the same 0.125 by 0.125 square region from the original digitized photograph. All three images would look identical from a distance. For each image, the 0.125 by 0.125 foot square patch was extended to a 1.0 by 1.0 foot square patch. The actual camouflage pattern to be printed on camouflage fabric is assumed to be an eight foot square. Since the camouflage image consists of a 1000 by 1000 pixel resolution, resolution of a primitive element printed on the camouflage fabric would

be approximately 0.1 inch square. Algorithms that produce each version of the pattern are explained below.

Image 3. This camouflage image is generated using linear interpolation and adaptive quantization. First, using linear interpolation the image is made 8 times larger. Then the resulting image is passed through a quantizer. The quantizer takes an image with a broad range of gray values and changes it to an image with a limited number of gray values. In this particular example we used only three gray values. The algorithm assigns the nearest gray value to a given pixel, and carries any resulting error in that value to adjacent pixels.

For this image we used three gray levels 256, 128 and 0 to quantize 256 original gray levels. The algorithm is given below:

1. Set error to zero.
2. Add next pixel's gray value to the error
3. If error is larger than 256, then set the pixel's gray value to 256 and subtract 256 from error value and go to 6
4. If error is bigger than 128, then set the pixel's gray value to 128 and subtract 128 from error value and go to 6
5. Set the pixel's gray value to 0.
6. If all the pixels are not processed, go to 2

We can easily manipulate this algorithm to utilize more than 3 gray levels. Quantizing gray levels can be chosen arbitrarily. In some cases instead of equally spaced gray values, it may be better to use the gray levels that are most predominant in the original image.

Images 4 and 5. These images illustrate different versions of the expansion algorithms. These algorithms are described in our May, June and July reports. In these algorithms a pixel is replaced by a square block with limited gray values (three in this example) such that the average gray value of the block is the same as the pixel's gray value. The block consists of two regions. Deterministic blocks exhibit regular replacement patterns for each possible gray shade as shown in figure 2.

Image 4.a was made by replacing every pixel with deterministic square blocks. Image 4.b uses the same algorithm but adds a probabilistic approach to the placement of the vertical stripes as shown in figure 3.

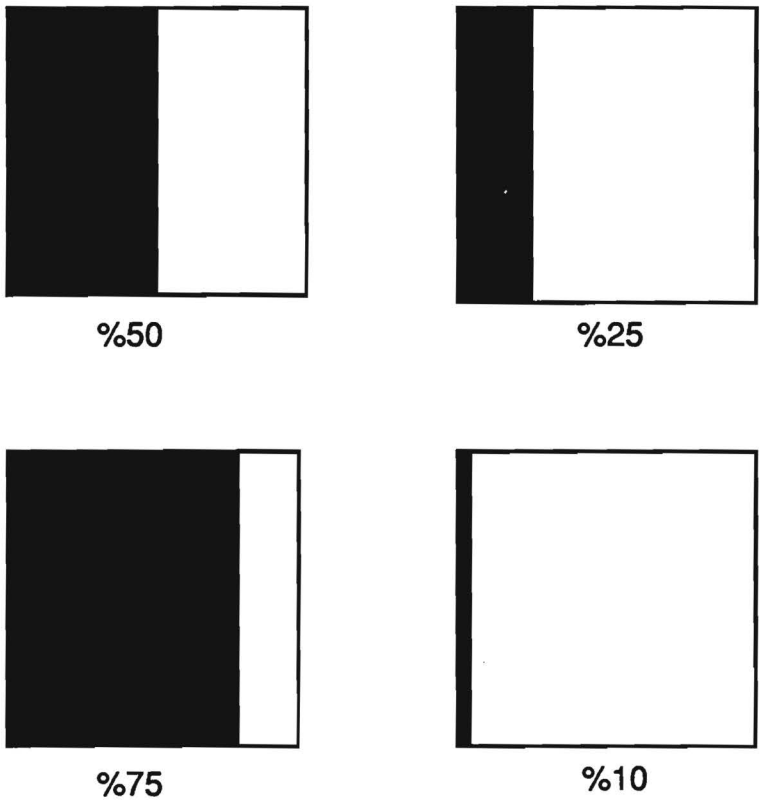


Figure 2. Deterministic square blocks

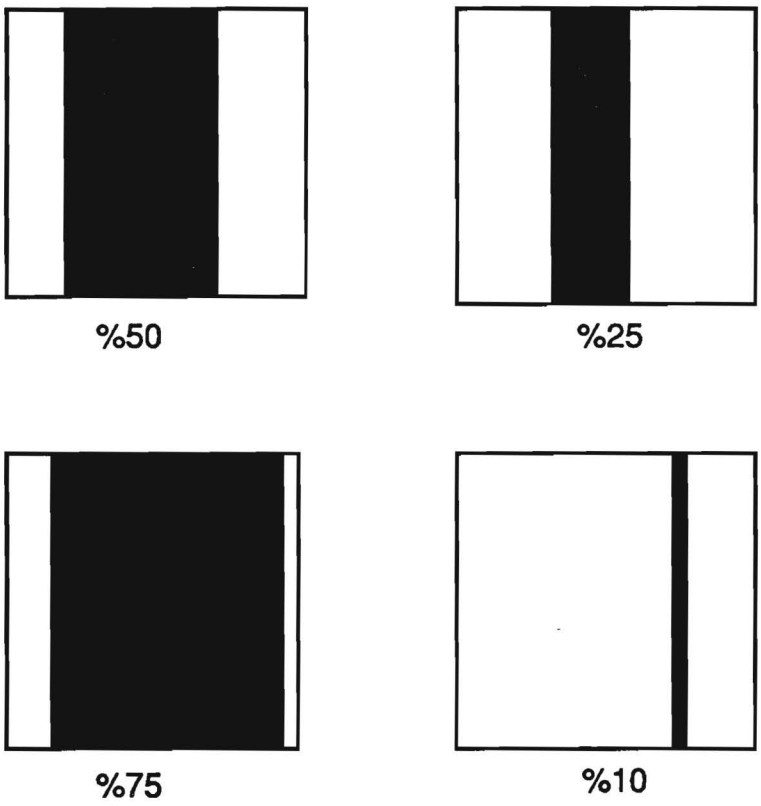


Figure 3. Deterministic shapes with probabilistic placing

Image 5 is a combination of fractal shapes and deterministic shapes with probabilistic placing. The fractal algorithms are explained in previous reports in May and June.

Images 6-8. These images illustrate the camouflage patterns on the original scene. Here the scene is observed from 256 feet. The photograph image is extended using linear interpolation and the camouflage image is average filtered down to 1/4 by 1/4 foot squares that appear in the scene.

Image 1

Image 2

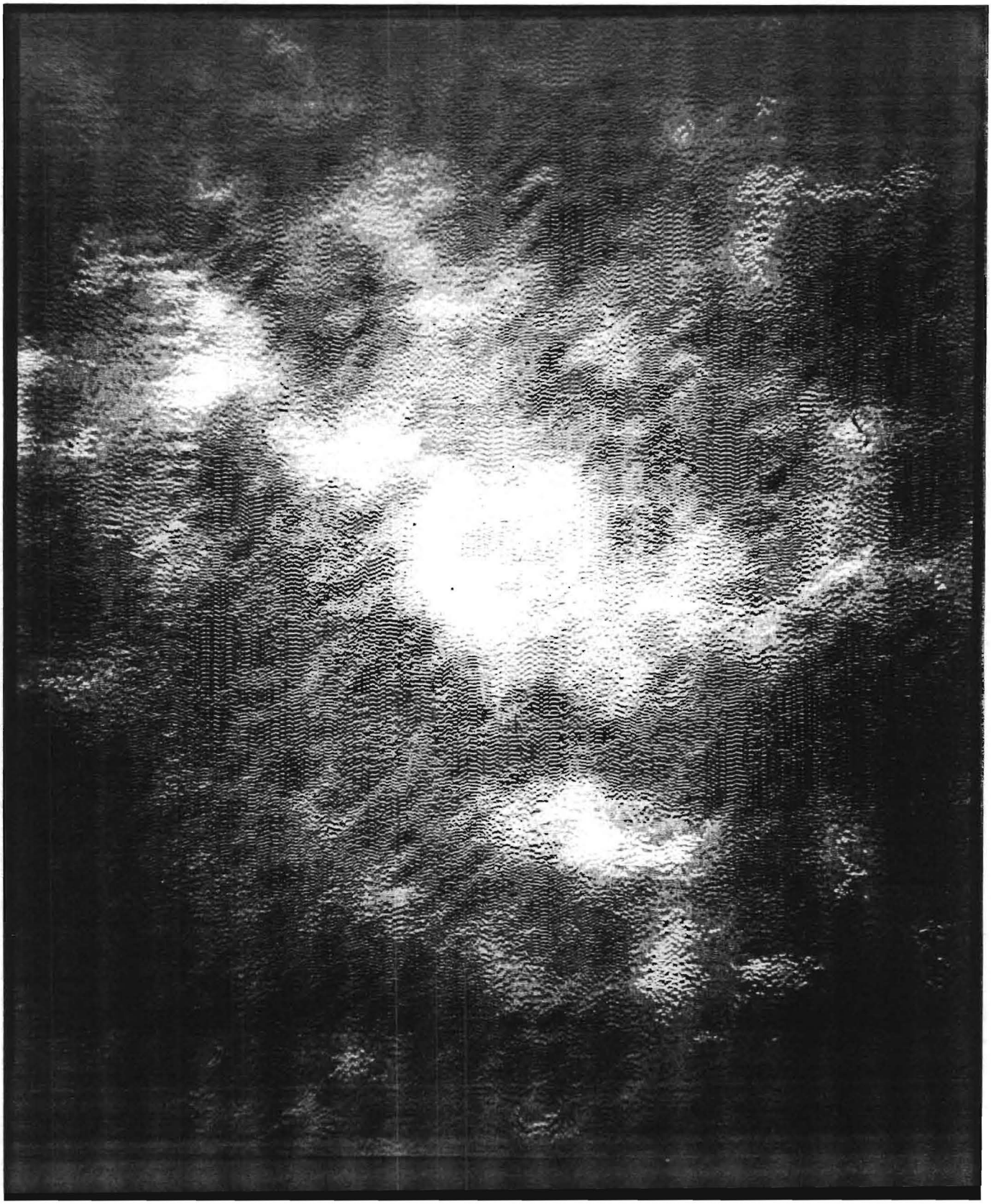


Image 3

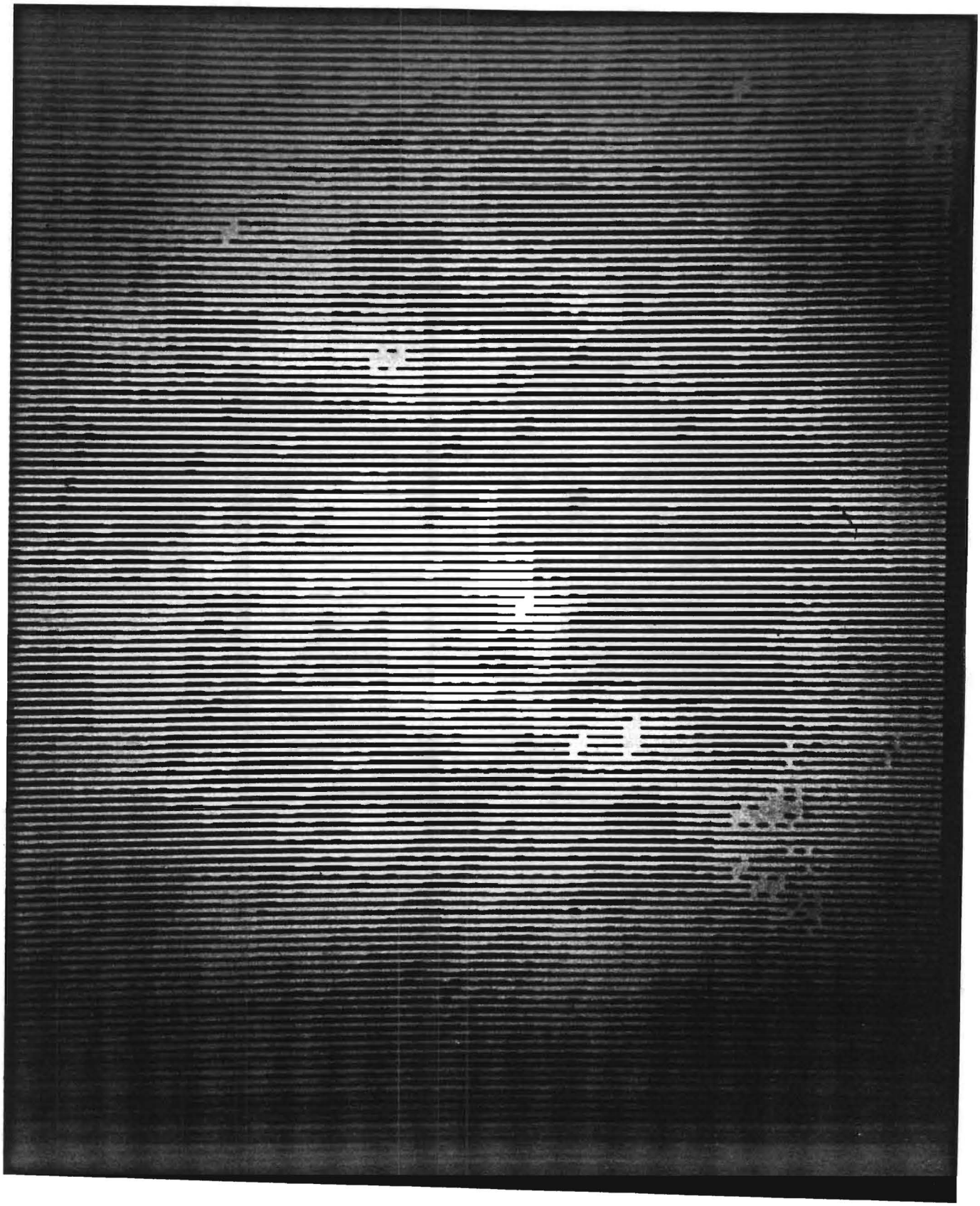


Image 4a

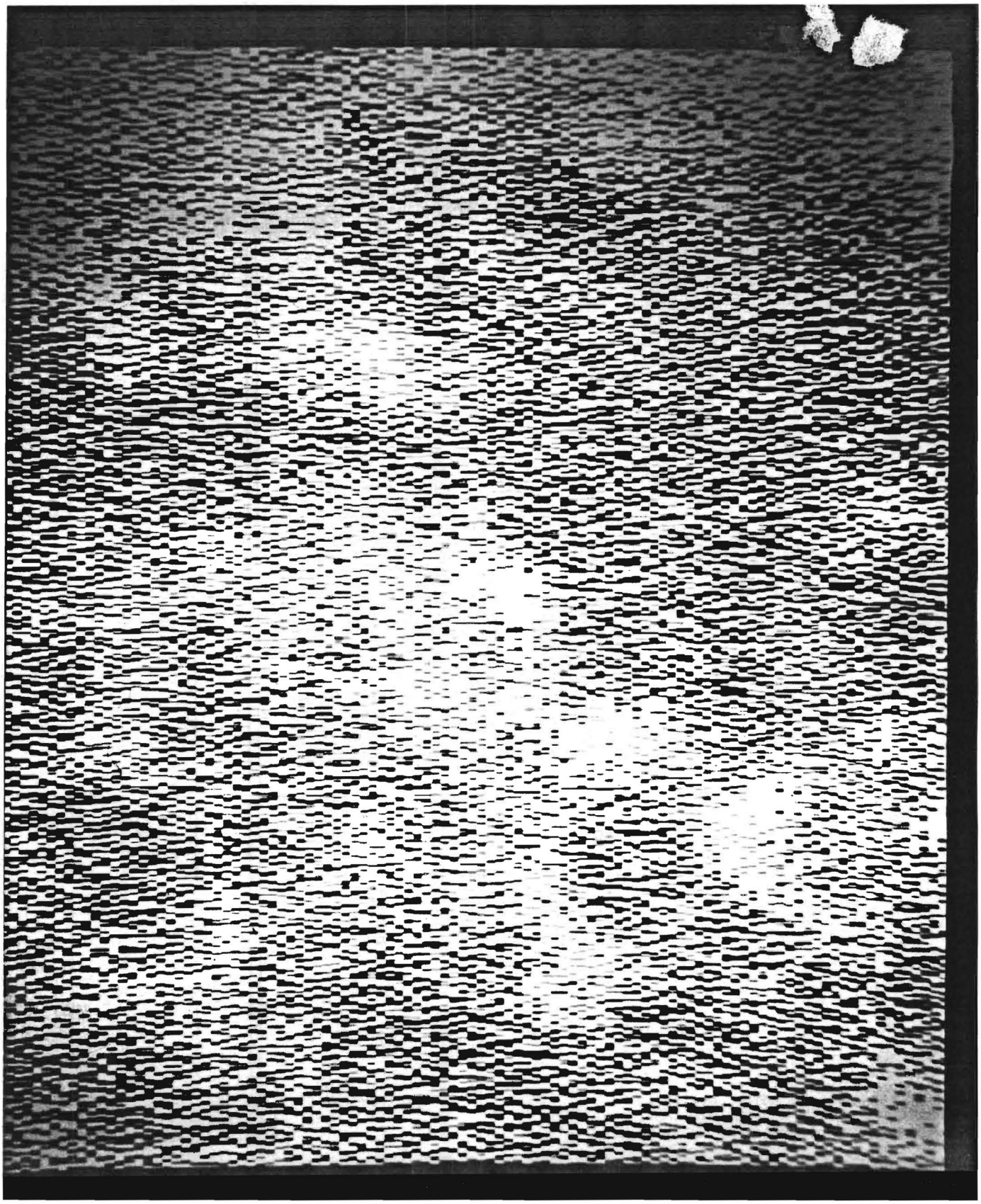


Image 4b

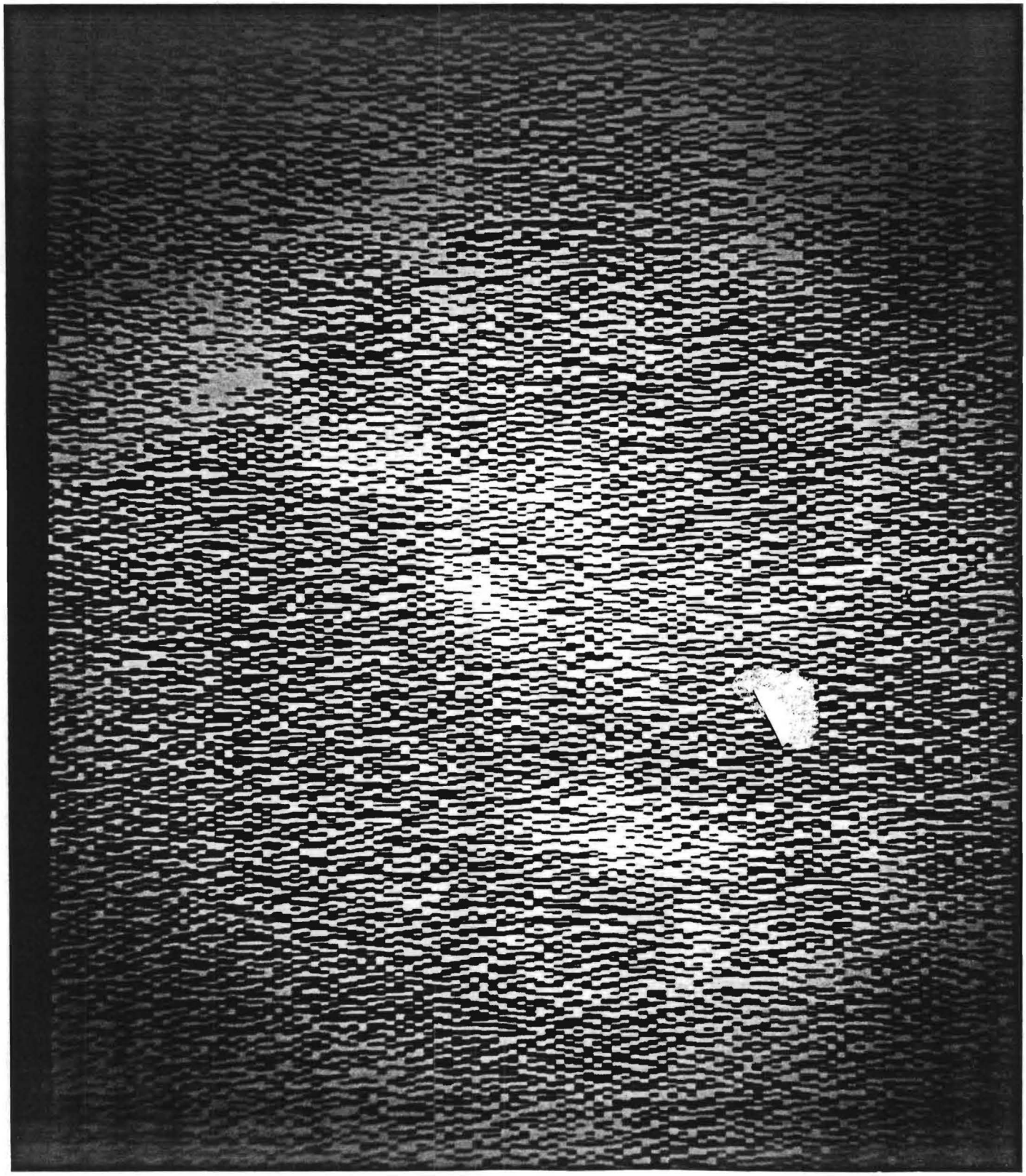


Image 5

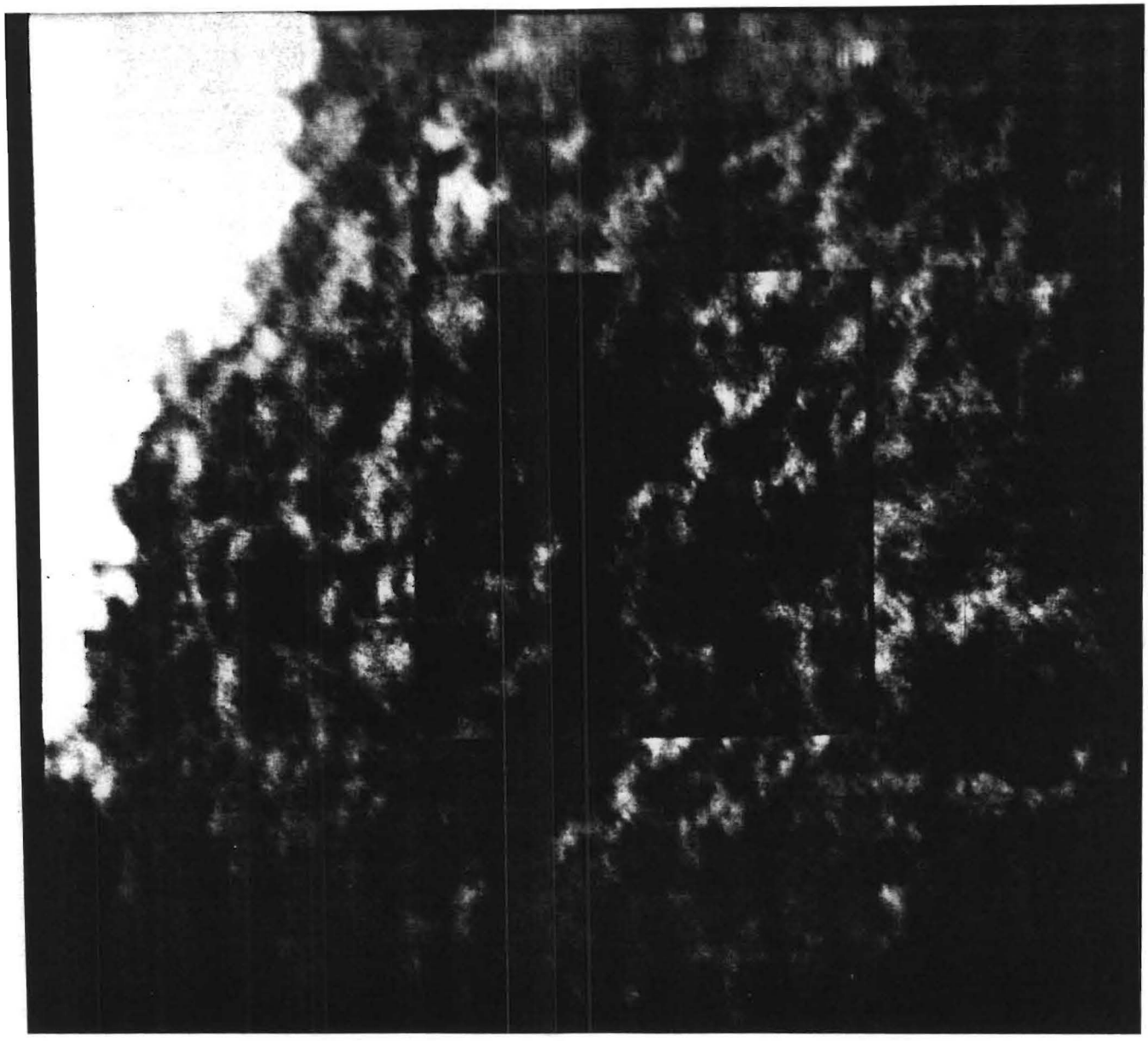


Image 6

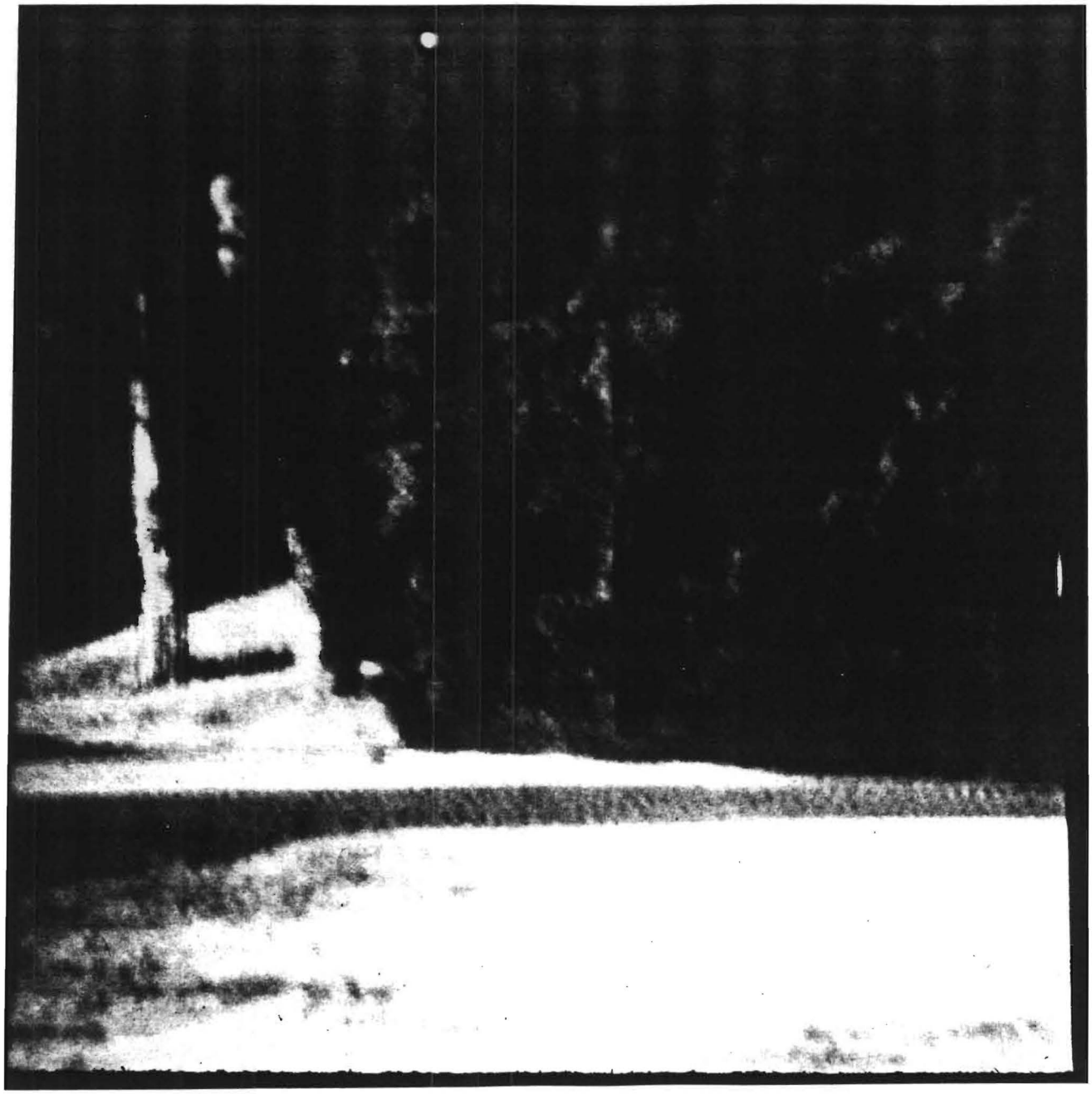


Image 7

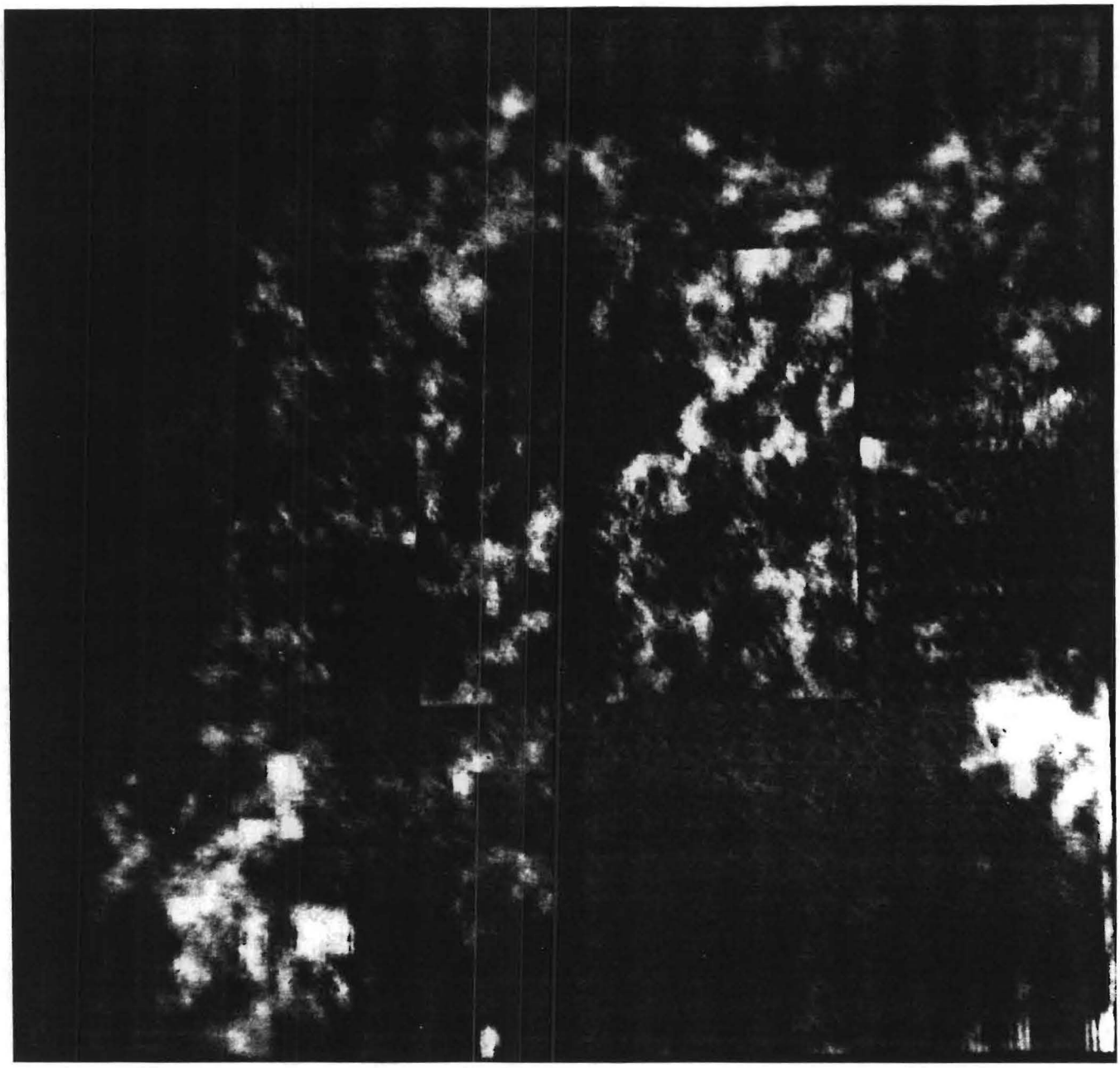


Image 8

PLANNING FOR PROTOTYPE SOFTWARE

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

October 15, 1990

Larry F. Hodges
Ergun Akleman

Planning For Prototype Software

During the past month we have been evaluating algorithms that were described in the previous reports as to their suitability for a prototype camouflage-generation software package. This planning included several phone conversations and a meeting with Mr. Jerry Edwards on September 14, 1990. We have also developed a digitizing process in our animation laboratory at Georgia Tech that will allow us to digitized color photographs of scenes. Until now we have been working with B&W digitized images from TBE. By our November report we intend to present an overview of our proposed prototype software system for review by Mr. Edwards.

CAMOUFLAGE TOOLKIT

MONTHLY REPORT
(Double Report)

Prepared for
TELEDYNE BROWN ENGINEERING

November 15, 1990
December 15, 1990

Larry F. Hodges
Ergun Akleman

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280

Camouflage Toolkit

As agreed upon by Mr. Jerry Edwards of TBE during our meeting on the Georgia Tech campus on December 13th, this report is a double report and will replace the two reports originally due on November 15 and December 15. During the past two months we have concentrated on building software routines to support a general camouflage generation program. These routines are called the *Camouflage Toolkit*. Currently an operator must first choose an area from a digitized image from which a camouflage pattern will be defined. The area is expanded and quantized. The resulting quantized image is then blown up in scale to a size that can be manufactured. Several different variations of a basic pattern can be created by specifying different intensities and color combinations.

Currently the camouflage toolkit consists of the following routines:

1. ***Quantize*** red green blue intensity imagename.RGB

This program will quantize a given image stored as a two-dimensional array of RGB integer triples in a file with name format *imagename*.RGB. Intensities of red, blue, and green of the original image are assumed to be in the range 0..255 for each base color. Quantization of the image will reduce the total number of intensities for each base color as desired by the operator. For instance red=1 green=2 blue=2 will quantize the image only using one shade of red and two shades of green and blue. The intensity parameter adjusts the overall intensity of the image. This feature will help to adjust the subjective qualities of the image.

2. ***Lowpass*** image1.RGB image2.RGB

This program filters image1.RGB. The resulting image image2.RGB is a reduced size lowpassed version of image1.RGB. This program will help to understand how image1.RGB will be seen from a distance. Image1.RGB and image2.RGB store an image as a logical two-dimensional array of RGB triples.

3. ***Linint*** image1.RGB image2.RGB:

This program linear interpolates image1.RGB to a larger size and stores the result in image2.RGB. Image1.RGB and image2.RGB store an image as a

logical two-dimensional array of RGB triples.

4. ***Thicken*** image1.RGB image2.RGB

This program transforms each pixel in image1.RGB to a 2x2 pixel square with the same color value and puts the resulting image in image2.RGB. Image1.RGB and image2.RGB store an image as a logical two-dimensional array of RGB triples.

5. ***Getsquare*** image1.RGB image2.RGB xmin ymin xmax ymax:

This program copies a square area defined by coordinates: xmin, ymin, xmax, ymax from an image stored in file image1.RGB. The square area is stored in file: image2.RGB. Image1.RGB and image2.RGB store an image as a logical two-dimensional array of RGB triples.

ADDITIONS TO CAMOUFLAGE

PATTERN GENERATION ROUTINES

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

January 15, 1991

Larry F. Hodges
Ergun Akleman

Additions to Camouflage Pattern Generation Routines

INTRODUCTION

Based on talks before Christmas break with Mr. Jerry Edwards we are expanding our planned software product for camouflage generation to provide a greater array of image generation procedures. In addition to the image processing methods that we have already developed and reported on we now plan to add computer graphics facilities for image modification and generation. In particular, these three facilities will be added:

1) Paint facilities
2) Draw facilities
3) Fractal related draw facilities

PAINT FACILITIES

In the standard terminology the name paint implies pixel by pixel drawing. The image is directly drawn on the screen. This facility is intended to allow the user to make minor adjustments on the final camouflage pattern.

DRAW FACILITIES

In draw mode, geometrical objects such as circles, polygons or lines, are generated and stored as separate objects that can be integrated into a camouflage pattern.

FRACTAL RELATED DRAW FACILITIES

In addition to the classical geometric objects we will add a fractal generation facility. We are planning to provide the user the ability to easily create fractals that resemble natural objects. A fractal object may also be combined with the other geometrical objects.

These simple facilities, in conjunction with our image processing algorithms, will allow the user simple procedures to modify or create patterns for camouflage generation.

Development of a Merit Function

To Quantify Camouflage Patterns

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

February 15, 1991

Larry F. Hodges
Ergun Akleman
Ted Doll

Development of a Merit Function

To Quantify Camouflage Patterns

Ted Doll, a Senior Research Scientist in the Electro-optics Laboratory of the Georgia Tech Research Institute, has joined the camouflage characterization and development team at Georgia Tech. Dr. Doll has done previous work in the testing of camouflage patterns and will contribute to the development of a merit function that quantifies the extent to which a camouflage pattern blends into a given background. There are a number of merit functions which quantify the complexity or "clutter" of an image. A routine could be coded that automatically calculates the clutter metric for both a selected camouflage pattern and a scene, or part of a scene, in which a target might plausibly be located. The difference between the values of the clutter metric for the pattern and the scene would be a measure of the conspicuity or conversely, the detectability, of the pattern in that scene.

A tool which allows TBE to compute a clutter number could be extremely useful for comparing the merits of different patterns in a given background, or across a range of different backgrounds. The tool might also allow the user to designate a portion of a scene to be used as the background. For example, this could allow the user to evaluate a camouflage pattern against a treeline and areas in front of the treeline and exclude sky and tops of trees from the comparison.

By defining patches of different size as the camouflage pattern, the tool would also allow the user to ask how heterogeneous a camouflage pattern should be. For example, a camouflage pattern could be based only on the pattern of leaves of trees, or on a patch that includes other textures, such as rocky or grassy areas as well as leaves.

Some clutter metrics, such as one developed at GT, are computed by moving a window over the scene. The GT clutter metric (the RMS of pixel intensities) is computed for each window and then averaged over the whole scene. The window is a square whose dimension is twice the maximum target dimension. This has the effect of a high-pass filter. The

clutter metric reflects only fundamental frequency of the target.

There are other measures of clutter that are sensitive to the size of the target, or to specific features within the target. Two other possible metrics are the correlation length or the Fourier transform of pixel intensities. Clutter metrics of this sort could be used to evaluate the detectability of specific target features, such as the gun barrel of tanks, with and without camouflage. The TB toolkit might include some way of combining clutter measures for a number of features of a target into an overall measure of its detectability.

Clutter metrics might also be extended to measure variations in chromaticity rather than just luminance. There is a well-established model of human color perception in which all possible chromaticities can be expressed as two-dimensional vectors. The magnitudes of these vectors could be measured with a colorimeter for all combinations of Red, Green, and Blue on the CRT. The color vector magnitudes could be substituted for pixel intensities in the formula for the clutter metric. This would produce a measure of the merit of a camouflage pattern in terms of color as well as of luminance.

Image Characterization Methodology

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

March 15, 1991

Larry F. Hodges, Assistant Professor
Ergun Akleman, GRA
Ted Doll, Senior Research Scientist

IMAGE CHARACTERIZATION METHODOLOGY

INTRODUCTION:

The goal of the methodology outlined in this report is to quantify the extent to which patterns look alike to a human observer. The methodology is designed to capture critical aspects of human pattern vision; however, it is as yet untested. It is recommended that tests with human observers be conducted to determine how well the metric predicts observer detection performance.

The metric is under consideration to be added to the "toolbox" in the Teledyne-Brown camouflage generation system to provide a convenient and quantitative way to evaluate the similarity of patterns from the standpoint of human observers.

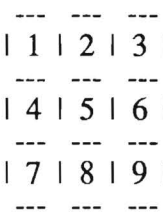
The user will be able to outline any area of an image to be evaluated. For example, one area might contain a candidate pattern for camouflage. Another area might include those locations in a scene where it is physically possible to locate a camouflage net (e.g., at the base of a treeline or in a grassy field, but not in the sky or in the tops of trees).

The methodology outlined here can be used to compute a set of image parameters for each of the areas to be evaluated. The parameters indicate the size, orientation, and consistency (in size and orientation) of the patterns in each area. The user will be able to evaluate the effectiveness of the camouflage pattern by comparing the values of the parameters for the two areas. The user will also be able to diagnose how the camouflage pattern could be improved by examining on which parameters the two image areas differ most.

It is important to have a methodology of this sort because a given camouflage pattern cannot match all backgrounds. It therefore becomes desirable to quantify how well a given camouflage pattern does over a variety of backgrounds. This methodology could be used to compute the average "match" of each of several different camouflage patterns to a set of backgrounds.

METHODOLOGY:

The image metric is computed by moving a "window" across the area of the image to be evaluated. The window has nine square cells of equal size, as illustrated below.



The window is moved over the area to be evaluated such that the center cell (#5) covers the entire area exactly once. At each position of the window, four values of the image metric, C_{ijk}, are computed using different combinations of cells. The subscript i represents the position of the window within the image area to be evaluated.

The metric is defined as the ratio of two variances:

$$C_{ijk} = Sc_{ik}^2/Ss_{ijk}^2$$

where Sc_{ijk}^2 is the variance of pixel luminances within the i th center cell, and Ss_{ijk}^2 is the variance between the average pixel luminances of two of the eight surrounding cells.

The subscript j indicates the orientation of the two surrounding cells used to calculate Ss_{ijk}^2. When j=0 degrees, the top and bottom cells (#s 2 and 8) are used; when j-45 degrees, cells 1 and 9 are used; and so on.

The subscript k is indicative of the cell size, and related to the pattern size that produces the largest ratio, C_{ijk2}. Each cell is a square whose sides are each 2k in length. k is the area that a pattern lying entirely within the center cell must have in order to produce the maximum value of the numerator, Sc_{ik}^2, of the ratio. The calculations are repeated for a range of different k values. The values of k used should bracket the size of the patterns in the area being evaluated.

The relative luminance of each RGB pixel combination can be determined by multiplying the spectral radiance function of each pixel by the luminous efficiency function for the CIE standard observer, and adding the three luminances (R, G, and B) for each pixel combination.

Sc_{ik}^2 is defined as follows:

$$Sc_{ik}^2 = \frac{1}{N-1}\left[\sum_{1=1}^{N} x_1^2 - \frac{1}{N}\left(\sum_{1=1}^{N} x_1\right)^2\right]$$

Where N is the number of pixels in the i th center cell of size k and x_1 is the luminance of the 1 th pixel in the ik th center cell. The subscripts i and k are omitted on the right side of the formula to simplify the presentation.

The variance of the means of the surrounding cells is:

$$Ss_{ijk}^2 = \left[\sum_{m=1}^{2} T_m^2/N\right] - T^2/kN$$

Where N is the number of pixels in each of the surrounding cells in direction j for the ik th window, T_m is the total of the pixel luminances for the m th surrounding cell in direction j of the ik th window, and T is the total luminance of all the pixels in the surrounding cells in direction j of the ik th window.

If there is no consistent periodicity to the pattern (i.e., the pattern is random noise), then the expected values of Sc_{ik} and Ss_{ijk}^2 are equal, and the expected value of C_{ijk} is I.O.

OUTPUT OF THE METHODOLOGY:

The metric, C_{ijk}, is designed to indicate the presence of a pattern of a specified size and orientation. An object that lies completely or mostly within the center cell and whose area is roughly one-half that of the center cell will make the numerator, and hence the ratio, large. Edges of larger objects that extend beyond the center cell in the direction of the surrounding cells, but still occupy about half the area of the center cell, contribute to both the numerator and denominator, and therefore tend to make the ratio smaller.

The major output is a table of average values of the metric over the image area for each orientation and cell size. The average value of the metric is designated as $C_{.jk}$. For each image area evaluated, these values will be tabled as a function of j and k, as illustrated below:

		j (degrees orientation)			
		0	45	90	135
	10				
	20		$C_{.jk}$ values		
k (pixels size)	30				
	40				
S_{js}:		S_0	S_{45}	S_{90}	S_{135}

Since the expected value of the C_{ijk}'s are 1.0 if the pattern is random noise, the expected values of the $C_{.jk}$'s are also 1.0 for a random noise pattern. To the extent that there is a consistent repetition of a pattern of a given size and orientation, some of the $C_{.jk}$ values will be greater than 1.0. The variance of the $C_{.jk}$'s, VAR ($C_{.jk}$), will be used as an indication of the consistency of the pattern.

The relative magnitudes of $C_{.jk}$ values will indicate the size and orientation of the pattern. For example, an elliptical pattern whose major axis is 40 pixels in length and oriented 45 degrees from the vertical would show a large value at (j,k) = (45, 40). If its minor axis were 10 pixels in length, it would also show a large $C_{.jk}$ value at (135, 10). It would also show large $C_{.jk}$ values at intermediate k values (i.e., 20, 30) for j = 0 and 90 degrees.

The S_js at the bottom of the above table are weighted averages of pattern sizes for each orientation. They indicate the dominant size of the pattern at each orientation, and are defined as follows:

$$S_j = \frac{\sum_k C_{.jk}\ k}{\sum_k C_{.jk}}$$

If the pattern is symmetrical about the four orientations tested, then the S_j's will be roughly equal. The variance of the S_j's is an indication of the asymmetry of the pattern or patterns in the image area. The relative magnitudes of the S_j's indicate the dominant orientation of the pattern.

The "tool" to incorporate in the Teledyne-Brown system will output the table of $C_{.jk}$ values, the VAR ($C_{,jk}$), the S_j values, and the VAR (S_j) for each image area being evaluated.

Implementation of Image Characterization Tool

MONTHLY REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

April 15, 1991

Larry F. Hodges, Assistant Professor
Ergun Akleman, GRA
Ted Doll, Senior Research Scientist

Implementation of Image Characterization Tool

Larry F. Hodges, Assistant Professor
Ergun Akleman, GRA
Ted Doll, Senior Research Scientist

Progress during the past thirty days has concentrated on developing a meaningful and user friendly visual representation of the clutter metric described in last month's report. This visual representation will be used to both understand and refine the metric and to present it eventually as a tool in the toolbox for creating and analyzing camouflage.

Currently the prototype program: CLUTDATA reads in a file and renders a file to be analyzed in a window on the CRT screen. The user can then interactively choose one of two different sampling methods from a menu. The clutter metric described in last month's report is then implemented and the results are displayed in three windows on the screen. One window displays raw values from the metric, the other two windows display two different visual representations of the graph of C/C_{max} values from the clutter metric.

Video Demonstration of Camouflage Patterns Generated From Natural Scenes

Consists of Video Tape and One-Page Overview

MIDTERM REPORT

Prepared for
TELEDYNE BROWN ENGINEERING

September 15, 1990

Larry F. Hodges
Ergun Akleman

Notes on Video Tape: Characterization and Generation of Camouflage Patterns

(Camouflage, the Video)

Larry F. Hodges, Ph.D.
Ergun Akleman

This video tape is intended to demonstrate the potential for creating camouflage patterns directly from an image block taken from a photograph of a natural scene. To begin, we cut an appropriate block from a photograph of an example scene. Our claim is, if we generate a camouflage pattern based on the information in this image block, it will blend into any similar natural scene.

To visually demonstrate this concept, we first take an image block and overlay multiple copies of it onto a scene. The blocks readily fade into the background. We have drawn a line at the bottom of each overlay so that they are easier to find. We then darken the background image to show how many overlays we have put into the scene.

Using the same image block , we then generate three camouflage patterns with different average intensity values. The different average intensity values simulate the actual shading and illumination effects that would occur if camouflage material with these patterns was put into real scenes. To generate the camouflage patterns we keep the texture information in the image block but reduce the total number of color intensities. For all three camouflage patterns shown we used only two shades of red, two shades of blue and three shades of green. We also reduce the size of the camouflage patterns by passing them through a low pass filter. This was done to reduce flutter in the video tape image since the NTSC signal's high frequency response is worse than that of the computer graphics workstation on which the images were originally created.

We then overlay the original image block and the three camouflage patterns onto the orginal scene. We choose the camouflage pattern whose average intensity value most closely matches the illumination in the scene and remove the other two and the original block image. Multiple copies of the camouflage pattern fade into the image to a remarkable degree. To show where the camouflage overlays are placed we animate them.

To illustrate that the camouflage pattern will also blend into scenes other than the one from which it was taken we demonstrate the same pattern with a different wooded scene and a rocky terrain scene. The fractal pattern characteristics of natural scenes which we have preserved in the camouflage cause the pattern to blend into the background with little sensitivity to placement or scale!